

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



PROYECTO FIN DE MÁSTER

**Técnicas de aprendizaje máquina para detección y
cuantificación de lluvia sobre imágenes de videovigilancia**

| | |
|-------------------|---|
| Titulación | Máster Universitario en Ingeniería de Telecomunicación |
| Autora | Naira Dolores Rosales Hernández |
| Tutores | Dr. Juan Luis Navarro Mesa Dr. Iván Daniel Guerra Moreno |
| Fecha | 9 de Julio de 2021 |

Agradecimientos

Me gustaría mostrar mi agradecimiento a mis tutores Dr. Juan Luis Navarro Mesa y al Dr. Iván D. Guerra Moreno por enseñarme y guiarme en el transcurso de este TFM. Gracias a la guía constante y orientación han conseguido mostrarme algo que era complejo en algo simple y accesible. Y, sobre todo, gracias al Dr. Juan Luis Navarro Mesa por las numerosas tutorías, por responder a cada email y WhatsApp enviado y por resolver todas las dudas. Ambos han conseguido que alcanzara un objetivo complejo de forma sencilla y clara.

También quiero agradecer a mi familia el apoyo incondicional, los ánimos y la fuerza que me transmiten cada día. Mis padres, hermanas, mis suegros... Y en especial a mi hija Adassa y mi pareja Nicanor por esas sonrisas cómplices que me llenan de felicidad cada día. Gracias por todo el amor que me dan cada día, que tanto necesito, y es mucho más de lo que me merezco.

Y, por último, agradecer al Sr. Prem Rawat por transmitirme su constante inspiración y pasión por disfrutar cada instante de esta vida, por enseñarme y mostrarme la fuente de la alegría en mi interior.

Resumen

Desde siempre los seres humanos han mostrado un gran interés en la meteorología para multitud de fines, como son el cuidado de los cultivos, el transporte, tanto aéreo como marítimo, o la protección de las personas ante desastres naturales, entre otros muchos casos. Por tanto, observar y predecir las condiciones meteorológicas es de importancia vital.

Poco a poco, mediante la incorporación de nuevas tecnologías, se realizan mejores estimaciones y predicciones del clima. Actualmente, la red de observación de la Organización Meteorológica Mundial (OMM) desarrolla proyectos para mejorar los sistemas de observación actuales. En los últimos años, OMM ha incorporado nuevas tecnologías, tales como, estaciones meteorológicas, radares meteorológicos, satélites meteorológicos, boyas meteorológicas, los buques de observación marítima, las radiosondas, entre otros.

El uso de estos sistemas conlleva un gran costo para su instalación, puesta en funcionamiento y mantenimiento. En este sentido, una de las ideas que subyace en este proyecto es la de avanzar hacia una ampliación de los sistemas de observación con un coste lo más bajo posible. Trabajaremos sobre la idea de dar valor añadido a sistemas de observación ya existentes. Es el caso de los sistemas de videovigilancia, basados en observación con cámaras IP, etc.

Este TFM es un proyecto que se basa en sistemas de videovigilancia meteorológica para el seguimiento de riesgos medioambientales.

Las líneas de trabajo que se desarrollarán están integradas en el Proyecto VIMETRI-MAC [VIMETRI-MAC], y es de potencial uso con cámaras de videovigilancia, propias y también públicas o privadas, desplegadas por las Islas Canarias para la detección de eventos de lluvia. Con este sistema se obtienen una serie de imágenes que se utilizarán para reconocimiento de patrones e identificar dichos eventos en dos fases: Detección y posterior cuantificación.

En términos generales, el valor añadido y novedoso que se aporta en este proyecto es ampliar la cobertura espacial y el muestreo temporal de un sistema de predicción meteorológica de lluvia mediante las imágenes captadas desde una red de cámaras de las estaciones de videovigilancia.

Como parte del núcleo principal se desarrollarán técnicas de procesado de señal y se hará uso de técnicas de Aprendizaje Máquina (Machine Learning y Deep Learning).

Abstract

1. Title of the Project

Machine learning techniques for rain detection and quantification from video surveillance images.

2. Abstract

Human beings have always been interested on the weather. To have a knowledge of the weather is important to agriculture, air and marine traffic, for instance. Nowadays, there is expensive technology that is used to predict the weather. An alternative could be to employ digital sensors for capturing images and then a method for weather detection can be used, including *machine learning*, such as Deep Learning. A methodology is presented to follow rain or no-rain conditions and it is a new instrument for detection and quantification of rain precipitation. This methodology consists of four main elements or modules. The first module is the sensor for acquiring the analog images and it generates digital images where the weather conditions are expected to be present. The second module performs feature extraction. In this module the digital images are transformed into a set of relevant characteristics, the feature sets. These sets are approximations of the acquired images but contain more discriminatory and invariant information than the raw digital data. The third module is the database module where the features extracted by the second module are stored. In the final module the system checks whether the extracted features have matching templates appropriate for rain detection and quantification. Finally, in this project we explore the intensive use of techniques based on machine learning with the aim of achieving automatic learning that allows us to achieve our fundamental objectives. One of the final results is a Toolbox performed with Matlab where the signal processing and machine learning techniques used will be gathered.

3.Keywords

Rain detection, rain quantification, weather observation, deep learning, machine learning, signal processing of images.

Índice General

| | |
|---|----|
| Agradecimientos | 6 |
| Resumen | 7 |
| Abstract | 8 |
| 1. Title of the Project | 8 |
| 2. Abstract..... | 8 |
| 3. Keywords | 9 |
| Índice General | 11 |
| Índice de Figuras | 14 |
| Índice de Tablas | 21 |
| Índice de Ilustraciones | 22 |
| Acrónimos..... | 23 |
| Capítulo 1: | 26 |
| 1. Introducción | 26 |
| 1.1 Antecedentes del trabajo..... | 26 |
| 1.2 Objetivos | 31 |
| Objetivo principal:..... | 31 |
| Objetivos secundarios:..... | 31 |
| 1.3 Estructura de la memoria | 32 |
| Capítulo 2: | 35 |
| 2. La precipitación..... | 35 |
| 2.1 Definición de precipitación | 35 |
| 2.2 Formación de la precipitación | 37 |
| 2.3 Clasificación de la precipitación | 38 |
| 2.3.1 Clasificación según las características físicas | 39 |
| 2.3.2 Clasificación según el mecanismo de formación | 39 |
| 2.3.3 Clasificación según la formación del meteoro | 42 |
| 2.3.4 Clasificación según la estructura | 45 |
| 2.4 Medidores de la precipitación actuales | 46 |
| 2.5 Estimación de la precipitación..... | 52 |
| 2.5.1 Intensidad | 52 |
| 2.5.2 Duración | 53 |
| 2.5.3 Frecuencia..... | 53 |
| Capítulo 3: | 55 |

| | | |
|-------------------|---|-----|
| 3. | La base de datos..... | 55 |
| 3.1 | Captación de los datos..... | 56 |
| 3.2 | Organización de los datos..... | 64 |
| Capítulo 4: | | 69 |
| 4. | Procesado y extracción de características de la imagen. | 69 |
| 4.1 | Cambios en el modo de color..... | 69 |
| 4.2 | Filtrado de corrección de blancos | 70 |
| 4.3 | Región de interés | 72 |
| 4.4 | Filtrado de gradiente. Algoritmo Sobel | 73 |
| Capítulo 5: | | 79 |
| 5. | Marco metodológico para la clasificación de las imágenes. | 79 |
| 5.1 | Generalidades del aprendizaje máquina | 80 |
| 5.2 | Formas de aprendizaje..... | 80 |
| 5.3 | Conceptos básicos de redes neuronales | 83 |
| 5.4 | Redes neuronales convolucionales..... | 87 |
| Capítulo 6: | | 93 |
| 6. | Método de detección y cuantificación de precipitación a partir de imágenes..... | 93 |
| 6.1 | Esquema general | 94 |
| 6.2 | Métricas de calidad | 110 |
| 6.2.1 | Parámetros de calidad para la detección | 110 |
| 6.2.2 | Parámetros de calidad para la cuantificación..... | 112 |
| 6.2.3 | Credibilidad de los parámetros de cuantificación | 114 |
| 6.3 | Criterios para realizar los experimentos..... | 119 |
| 6.4 | Experimentos de detección y cuantificación. Resultados..... | 121 |
| 6.4.1 | Experimentos con Luminancia..... | 122 |
| 6.4.2 | Experimentos con Cielab..... | 123 |
| 6.4.3 | Experimentos con balance de blancos | 125 |
| 6.4.4 | Experimentos con Sobel | 127 |
| 6.4.5 | Experimentos sobre la profundidad de la red neuronal..... | 131 |
| 6.4.6 | Experimentos sobre cuantificación | 142 |
| 6.4.7 | Resultados de los intervalos de confianza..... | 148 |
| Capítulo 7: | | 151 |
| 7.1 | Conclusiones | 151 |
| 7.2 | Líneas futuras | 153 |

| | |
|---|-----|
| Bibliografía..... | 224 |
| Pliego de condiciones..... | 229 |
| Pliego de condiciones de los recursos utilizados..... | 229 |
| Recursos hardware..... | 229 |
| Recursos software..... | 236 |
| Pliego de condiciones legales y administrativas..... | 237 |

Índice de Figuras

| | |
|--|----|
| IMAGEN 1. SISTEMAS DE OBSERVACIÓN METEOROLÓGICAS [OMM 2019]. | 27 |
| IMAGEN 2. RESULTADO DEL ALGORITMO DE DETECCIÓN DE LLUVIA [XUEYANG 2017]. | 28 |
| IMAGEN 3. TÉCNICA DE PROCESADO DE SEÑAL | 29 |
| IMAGEN 4. TÉCNICA DE SEGUIMIENTO DE LA LÍNEA DEL HORIZONTE PROBADA BAJO UNA VARIEDAD DE CONDICIONES DE ILUMINACIÓN Y ESTADO DEL MAR. EN LA ÚLTIMA IMAGEN SE REPRESENTA LA ESTABILIZACIÓN DE LAS DIFERENTES IMÁGENES CAPTADAS [SCHWENDEMAN 2015]. | 30 |
| IMAGEN 5. CICLO HIDROLÓGICO [SOCIEDAD GEOGRÁFICA DE LIMA 2011]. | 37 |
| IMAGEN 6. PRECIPITACIÓN CONVECTIVA. | 40 |
| IMAGEN 7. FORMACIÓN DE NUBES EN PRECIPITACIONES CONVECTIVAS. | 40 |
| IMAGEN 8. PRECIPITACIÓN OROGRÁFICA. | 41 |
| IMAGEN 9. FRENTE FRÍO. | 41 |
| IMAGEN 10. FRENTE CÁLIDO. | 42 |
| IMAGEN 11. MODELO DE GUNN Y KINZER. | 43 |
| IMAGEN 12. DIÁMETRO DE LA GOTA DE LLUVIA. | 43 |
| IMAGEN 13. FORMA DE LA GOTA DE LA PRECIPITACIÓN. | 44 |
| IMAGEN 14. ÁNGULO DE INCLINACIÓN SEGÚN EL RADIO DE LA GOTA DE LA PRECIPITACIÓN. | 45 |
| IMAGEN 15. EN LA IZQUIERDA, PLUVIÓMETRO CON EMBUDO PARA AUMENTAR EL ÁREA DE CAPTACIÓN DEL AGUA DE LA PRECIPITACIÓN. EN LA DERECHA, UBICACIÓN DEL PLUVIÓMETRO ESTABLECIDA POR LA NORMA EN ESPAÑA. [USAL 2020] | 47 |
| IMAGEN 16. LOS PLUVIÓMETROS HELLMANN ESTABLECIDOS EN EL ENTORNO DEL OBSERVATORIO ATMOSFÉRICO DE IZAÑA, EN EL PARQUE NACIONAL DE LAS CAÑADAS DE TEIDE [AEMET 2009]. | 48 |
| IMAGEN 17. EN LA IZQUIERDA EL PLUVIÓMETRO DE MEDIDOR DE FLOTADOR. EN EL CENTRO EL PLUVIÓMETRO DE MEDIDO DE BALANZA. EN LA DERECHA EL PLUVIÓMETRO DE CUBETA BASCULANTE. | 49 |
| IMAGEN 18. PLUVIÓMETRO DIGITAL. | 50 |
| IMAGEN 19. A LA IZQUIERDA, PLUVIÓMETROS VAISALA WXT530. | 51 |
| IMAGEN 20. CÁMARA MODELO GRANDSTREAM GXV364_FHD. | 57 |
| IMAGEN 21. CÁMARAS GXV364_FHD DE LA BASE | 57 |
| IMAGEN 22. VISIÓN DE LAS CÁMARAS GXV364_FHD DE LA BASE | 58 |
| IMAGEN 23. CÁMARAS GXV364_FHD DE LA BASE ULPGC. EDIFICIO DE ELECTRÓNICA Y TELECOMUNICACIÓN, EN LA CUBIERTA DEL PABELLÓN B. | 58 |
| IMAGEN 24. VISIÓN DE LAS CÁMARAS GXV364_FHD DE LA BASE ULPGC. EDIFICIO DE ELECTRÓNICA Y TELECOMUNICACIÓN, EN LA CUBIERTA DEL PABELLÓN B. | 59 |
| IMAGEN 25. CÁMARAS FOSCAM DE LA BASE ULPGC. EDIFICIO DE ELECTRÓNICA Y TELECOMUNICACIÓN, EN LA CUBIERTA DEL PABELLÓN B. | 61 |
| IMAGEN 26. VISIÓN DE LAS CÁMARAS FOSCAM DE LA BASE ULPGC. EDIFICIO DE ELECTRÓNICA Y TELECOMUNICACIÓN, EN LA CUBIERTA DEL PABELLÓN B. | 61 |
| IMAGEN 27. CONTENIDO DE LA BASE DE DATOS. | 64 |

| | |
|---|----|
| IMAGEN 28. SUBCARPETAS PARA CADA CARPETA DE CADA CÁMARA DE FORMA INDIVIDUAL. | 65 |
| IMAGEN 29. CONTENIDO DE LAS SUBCARPETAS DE LA CÁMARA HOSPITAL1-BARRANCO SECO. | 65 |
| IMAGEN 30. DIAGRAMA DE BLOQUES DEL PRE-PROCESADO DE LA SEÑAL. | 69 |
| IMAGEN 31. CAPTURA DE LAS IMÁGENES DESDE DIFERENTES CÁMARAS. | 70 |
| IMAGEN 32. NORMALIZACIÓN DE LA IMAGEN. | 70 |
| IMAGEN 33. IMÁGENES TOMADAS DESDE LA MISMA CÁMARA EN MOMENTOS DIFERENTES DEL DÍA. COMO SE OBSERVA HAY UN CAMBIO CONSTANTE DE LA TEMPERATURA DE COLOR DE LA IMAGEN. | 71 |
| IMAGEN 34. DIAGRAMA DE BLOQUES DEL BALANCE DE BLANCOS. | 72 |
| IMAGEN 35. POSIBLES REGIONES DE INTERÉS EN LAS QUE HEMOS DIVIDIDO LA IMAGEN. ESTAS IMÁGENES PERTENECEN A LA CÁMARA TAFIRA3-FOSCAM Y ROI NÚMERO 8. | 73 |
| IMAGEN 36. ZOOM DE LOS PÍXELES QUE FORMA LA IMAGEN. | 74 |
| IMAGEN 37. GRADO DE OSCURIDAD DE UN PIXEL. | 74 |
| IMAGEN 38. EJEMPLO DE LOS GRADIENTES DE LA IMAGEN. | 74 |
| IMAGEN 39. GRADIENTE EN UN PUNTO O PÍXEL. | 75 |
| IMAGEN 40. MÁSCARAS DE CONVOLUCIÓN DEL OPERADOR SOBEL PARA EL GRADIENTE FILA (MX) Y EL GRADIENTE COLUMNA (MY). | 76 |
| IMAGEN 41. GRADIENTE FILA Y COLUMNA APLICANDO EL VALOR DE $K=2$ DEL OPERADOR SOBEL. | 76 |
| IMAGEN 42. CÓDIGO DE MATLAB DONDE REALIZAMOS LA DOBLE CONVOLUCIÓN. | 76 |
| IMAGEN 43. MÁSCARA DIFERENCIAL Y MÁSCARA DE ALISAMIENTO. | 77 |
| IMAGEN 44. EN ESTA IMAGEN SE ILUSTRAN LAS IMÁGENES ANTES Y DESPUÉS DEL FILTRO SOBEL. PERTENECEN A LA CÁMARA TAFIRA3-FOSCAM Y ROI NÚMERO 8. INTRODUCIMOS EN LA PARTE SUPERIOR UNA IMAGEN EN BLANCO Y NEGRO Y DEBAJO LA MISMA IMAGEN A COLOR. | 77 |
| IMAGEN 45. DIAGRAMA DE BLOQUES DEL APRENDIZAJE SUPERVISADO. | 81 |
| IMAGEN 46. MODELOS DEL APRENDIZAJE SUPERVISADO. | 82 |
| IMAGEN 47. MODELOS DEL APRENDIZAJE NO SUPERVISADO. | 82 |
| IMAGEN 48. MODELOS DEL APRENDIZAJE POR REFUERZO. | 83 |
| IMAGEN 49. ESTRUCTURA DE UNA NEURONA. | 84 |
| IMAGEN 50. PRIMER MODELO MATEMÁTICO DE UNA NEURONA ARTIFICIAL. | 85 |
| IMAGEN 51. ENTRADAS Y SALIDAS DE UNA NEURONA QUE FORMA LA RED NEURONAL. | 85 |
| IMAGEN 52. RED NEURONAL. | 86 |
| IMAGEN 53. ESTRUCTURA DE UNA RED NEURONAL CONVOLUCIONAL. | 88 |
| IMAGEN 54. CAPA CONVOLUCIONAL. | 88 |
| IMAGEN 55. PROCESO DE FILTRADO DE LA CAPA CONVOLUCIONAL. | 89 |
| IMAGEN 56. RECORRIDO DEL FILTRO EN LA IMAGEN DE ENTRADA DE LA CAPA CONVOLUCIONAL. | 90 |
| IMAGEN 57. ALGORITMO MAX-POOLING. | 90 |
| IMAGEN 58. ALGORITMO AVERAGE-POOLING. | 90 |
| IMAGEN 59. ESQUEMA GENERAL DEL TFM. | 94 |

| | |
|---|-----|
| IMAGEN 60. ESQUEMA GENERAL DEL TFM CON UNA ENUMERCIÓN EN CADA PASO. | 94 |
| IMAGEN 61. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA. | 95 |
| IMAGEN 62. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA. | 95 |
| IMAGEN 63. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR LUMINANCIA. | 96 |
| IMAGEN 64. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR LUMINANCIA. | 96 |
| IMAGEN 65. ESQUEMA GENERAL DEL MODO DE COLOR DE LA LUMINANCIA. | 97 |
| IMAGEN 66. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR CIELAB. | 97 |
| IMAGEN 67. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR CIELAB. | 98 |
| IMAGEN 68. ESQUEMA GENERAL DEL MODO DE COLOR CIELAB. | 98 |
| IMAGEN 69. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR CIELAB DESPUÉS DE APLICARLE EL FILTRO DE BALANCE DE BLANCOS. | 99 |
| IMAGEN 70. IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR CIELAB DESPUÉS DE APLICARLE EL FILTRO DE BALANCE DE BLANCOS. | 99 |
| IMAGEN 71. ESQUEMA GENERAL DEL MODO DE COLOR CIELAB ANTES Y DESPUÉS DEL FILTRO DE BALANCE DE BLANCOS. | 100 |
| IMAGEN 72. REGIÓN DE INTERÉS DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR LUMINANCIA. | 101 |
| IMAGEN 73. REGIÓN DE INTERÉS DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR LUMINANCIA. | 101 |
| IMAGEN 74. ESQUEMA GENERAL DEL MODO DE COLOR DE LA LUMINANCIA ANTES Y DESPUÉS DE LA EXTRACCIÓN DE LA REGIÓN DE INTERÉS. | 102 |
| IMAGEN 75. REGIÓN DE INTERÉS DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR CIELAB. | 102 |
| IMAGEN 76. REGIÓN DE INTERÉS DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR CIELAB. | 103 |
| IMAGEN 77. ESQUEMA GENERAL DEL MODO DE COLOR DE LA CIELAB ANTES Y DESPUÉS DE LA EXTRACCIÓN DE LA REGIÓN DE INTERÉS. | 103 |
| IMAGEN 78. FILTRADO SOBEL DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR LUMINANCIA. | 104 |
| IMAGEN 79. FILTRADO SOBEL DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR LUMINANCIA. | 104 |
| IMAGEN 80. ESQUEMA GENERAL DEL MODO DE COLOR DE LA LUMINANCIA ANTES Y DESPUÉS DEL FILTRO SOBEL. | 105 |
| IMAGEN 81. FILTRADO SOBEL DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y SIN LLUVIA CON EL MODO DE COLOR CIELAB. | 105 |
| IMAGEN 82. FILTRADO SOBEL DE LA IMAGEN DE LA CÁMARA FOSCAM DE DÍA Y CON LLUVIA CON EL MODO DE COLOR CIELAB. | 106 |
| IMAGEN 83. ESQUEMA GENERAL DEL MODO DE COLOR CIELAB ANTES Y DESPUÉS DEL FILTRO SOBEL. | 106 |

| | |
|--|-----|
| IMAGEN 84. RESULTADO DEL ENTRENAMIENTO Y TEST DEL MODO DE COLOR CIELAB. | 107 |
| IMAGEN 85. RESULTADO DEL ENTRENAMIENTO Y TEST DEL MODO DE COLOR LUMINANCIA. | 107 |
| IMAGEN 86. EJEMPLO DE MÉTRICAS DE CALIDAD ALMACENADA. | 108 |
| IMAGEN 87. TABLA DE LOS DATOS OBTENIDOS CON EL MODO DE COLOR LUMINANCIA. | 109 |
| IMAGEN 88. TABLA DE LOS DATOS OBTENIDOS CON EL MODO DE COLOR CIELAB. | 109 |
| IMAGEN 89. EJEMPLO DE UN INTERVALO DE CONFIANZA AL 95%. | 115 |
| IMAGEN 90. FILTRADO DE MODO DE COLOR A LUMINANCIA. | 122 |
| IMAGEN 91. IMÁGENES ANTES Y DESPUÉS DEL FILTRADO DE LUMINANCIA. | 122 |
| IMAGEN 92. IMAGEN DE LA CÁMARA FOSCAM DURANTE LA NOCHE Y CON LLUVIA ALTA. | 123 |
| IMAGEN 93. IMAGEN DE LA CÁMARA HOSPITAL 3- LA ISLETA DURANTE EL DÍA Y CON LLUVIA ALTA. | 123 |
| IMAGEN 94. FILTRADO DE MODO DE COLOR A CIELAB. | 124 |
| IMAGEN 95. IMÁGENES ANTES Y DESPUÉS DEL FILTRADO DE CIELAB. | 124 |
| IMAGEN 96. IMAGEN DE LA CÁMARA FOSCAM DURANTE LA NOCHE Y CON LLUVIA ALTA. | 125 |
| IMAGEN 97. IMAGEN DE LA CÁMARA HOSPITAL 3- LA ISLETA | 125 |
| IMAGEN 98. BALANCE DE BLANCOS. | 126 |
| IMAGEN 99. IMÁGENES ANTES Y DESPUÉS DEL FILTRADO DE BALANCE DE BLANCOS. | 126 |
| IMAGEN 100. IMAGEN DE LA CÁMARA FOSCAM DURANTE LA NOCHE Y CON LLUVIA ALTA. | 126 |
| IMAGEN 101. IMAGEN DE LA CÁMARA HOSPITAL 3- LA ISLETA DURANTE EL DÍA Y CON LLUVIA ALTA. | 127 |
| IMAGEN 102. FILTRO SOBEL. | 127 |
| IMAGEN 103. IMAGEN DE LUMINANCIA Y CIELAB, ANTES Y DESPUÉS, DEL FILTRADO DE SOBEL SIN LLUVIA. | 128 |
| IMAGEN 104. IMAGEN DE LUMINANCIA Y CIELAB, ANTES Y DESPUÉS, DEL FILTRADO DE SOBEL CON LLUVIA. | 128 |
| IMAGEN 105. RESULTADOS OBTENIDOS SIN EL USO DEL FILTRO SOBEL CON EL MODO DE COLOR DE CIELAB. | 129 |
| IMAGEN 106. RESULTADOS OBTENIDOS CON EL USO DEL FILTRO SOBEL CON EL MODO DE COLOR DE CIELAB. | 129 |
| IMAGEN 107. RESULTADOS OBTENIDOS SIN EL USO DEL FILTRO SOBEL CON EL MODO DE COLOR DE LUMINANCIA. | 130 |
| IMAGEN 108. RESULTADOS OBTENIDOS CON EL USO DEL FILTRO SOBEL CON EL MODO DE COLOR DE LUMINANCIA. | 130 |
| IMAGEN 109. DIAGRAMA DE BLOQUES DEL SISTEMA DE DETECCIÓN INCLUYENDO LA ETAPA DE LA RED NEURONAL CONVOLUCIONAL. | 132 |
| IMAGEN 110. CÁMARA HOSPITAL1-BARRANCOSECO. | 134 |
| IMAGEN 111. CÁMARA HOSPITAL2-CATEDRAL. | 134 |

| | |
|--|-----|
| IMAGEN 112. CÁMARA HOSPITAL3-ISLETA. | 135 |
| IMAGEN 113. CÁMARA HOSPITAL4-ESCALERITAS. | 135 |
| IMAGEN 114. CÁMARA TAFIRA1-INFORMÁTICA. | 136 |
| IMAGEN 115. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 136 |
| IMAGEN 116. CÁMARA TAFIRA3-FOSCAM. | 137 |
| IMAGEN 117. CÁMARA TAFIRA4-FOSCAM. | 137 |
| IMAGEN 118. CÁMARA HOSPITAL1-BARRANCOSECO. | 138 |
| IMAGEN 119. CÁMARA HOSPITAL2-CATEDRAL. | 139 |
| IMAGEN 120. CÁMARA HOSPITAL3-ISLETA. | 139 |
| IMAGEN 121. CÁMARA HOSPITAL4-ESCALERITAS. | 140 |
| IMAGEN 122. CÁMARA TAFIRA1-INFORMÁTICA. | 140 |
| IMAGEN 123. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 141 |
| IMAGEN 124. CÁMARA TAFIRA3-FOSCAM. | 141 |
| IMAGEN 125. CÁMARA TAFIRA4-FOSCAM. | 142 |
| IMAGEN 126. DIAGRAMA DE BLOQUES INCLUYENDO LA ETAPA DE LA RED NEURONAL CONVOLUCIONAL. | 143 |
| IMAGEN 127. CÁMARA HOSPITAL1-BARRANCOSECO. | 144 |
| IMAGEN 128. CÁMARA HOSPITAL2-CATEDRAL. | 144 |
| IMAGEN 129. CÁMARA HOSPITAL3-ISLETA. | 145 |
| IMAGEN 130. CÁMARA HOSPITAL4-ESCALERITAS. | 145 |
| IMAGEN 131. CÁMARA TAFIRA1-INFORMÁTICA. | 146 |
| IMAGEN 132. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 146 |
| IMAGEN 133. CÁMARA TAFIRA3-FOSCAM. | 147 |
| IMAGEN 134. FRAGMENTO 1 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 154 |
| IMAGEN 135. FUNCIÓN DEL FILTRADO DE SOBEL. | 155 |
| IMAGEN 136. FRAGMENTO 2 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 156 |
| IMAGEN 137. EXCEL MATRIZ DE CONFUSIÓN. | 156 |
| IMAGEN 138. FRAGMENTO 4 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 157 |
| IMAGEN 139. ACCESO AL DIRECTORIO DE LOS VÍDEOS DE LAS CÁMARAS. | 157 |
| IMAGEN 140. FICHERO TXT CON LAS CÁMARAS USADAS EN EL TFM. | 158 |
| IMAGEN 141. LECTURA DE FICHERO TXT ANTERIOR. | 158 |
| IMAGEN 142. VALOR DE LAS FILAS Y LAS COLUMNAS. | 159 |
| IMAGEN 143. DIRECTORIOS LOS VÍDEOS DE CADA CÁMARA. | 159 |
| IMAGEN 144. FRAGMENTO 4 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 160 |
| IMAGEN 145. EXTRACCIÓN DEL NOMBRE DE CADA CÁMARA. | 160 |
| IMAGEN 146. DIRECTORIO DE LAS SUBCARPETAS DE CADA CÁMARA. | 161 |
| IMAGEN 147. VARIABLE DD REPRESENTA EL TAMAÑO DE LA TABLA. | 161 |
| IMAGEN 148. FRAGMENTO 5 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 161 |
| IMAGEN 149. ACCESO A DIRECTORIOS DE LOS VÍDEOS. | 162 |
| IMAGEN 150. DIRECTORIO DE CADA VIDEO. | 162 |
| IMAGEN 151. VALOR INICIAL DE LAS VARIABLES ID2 Y JD2. | 163 |
| IMAGEN 152. FRAGMENTO 6 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 163 |
| IMAGEN 153. FORMACIÓN DE UNA CADENA CON EL NOMBRE DEL VÍDEO. | 163 |
| IMAGEN 154. ACCESO A UN VÍDEO. | 164 |

| | |
|--|-----|
| IMAGEN 155. EXTRACCIÓN DE LAS PROPIEDADES DE UN VÍDEO. | 164 |
| IMAGEN 156. FRAGMENTO 7 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 165 |
| IMAGEN 157. FRAGMENTO 8 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 165 |
| IMAGEN 158. FRAGMENTO 9 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 166 |
| IMAGEN 159. FILTRADO DE MODO DE COLOR. | 166 |
| IMAGEN 160. EJEMPLO DE ALMACENAMIENTO DE LOS VÍDEOS DE LAS CÁMARAS. | 167 |
| IMAGEN 161. ASIGNACIÓN DE NOMBRE DE CADA IMAGEN TRAS PASAR EL PRIMER PREPROCESADO. | 168 |
| IMAGEN 162. FRAGMENTO 10 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 169 |
| IMAGEN 163. SELECCIÓN DE LAS REGIONES DE INTERÉS. | 170 |
| IMAGEN 164. DIVISIÓN DE LA IMAGEN EN DIFERENTES REGIONES. | 170 |
| IMAGEN 165. FRAGMENTO 11 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 171 |
| IMAGEN 166. DIRECTORIO DONDE SE ALMACENAN LAS IMÁGENES DE LUMINANCIA. | 171 |
| IMAGEN 167. NOMBRE CON EL QUE SE VAN ALMACENANDO LAS IMÁGENES. | 172 |
| IMAGEN 168. FRAGMENTO 12 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 173 |
| IMAGEN 169. DIRECTORIO DONDE SE VAN ALMACENANDO LAS IMÁGENES TRAS PASAR POR EL FILTRADO DE SOBEL. | 174 |
| IMAGEN 170. NOMBRE CON EL QUE SE ALMACENAN LAS IMÁGENES | 175 |
| IMAGEN 171. FRAGMENTO 13 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 175 |
| IMAGEN 172. FILTRO DE BALANCE DE BLANCOS. | 176 |
| IMAGEN 173. FUNCIÓN DE BALANCE DE BLANCOS. | 177 |
| IMAGEN 174. DIRECTORIO DEL BALANCE DE BLANCOS. | 178 |
| IMAGEN 175. FRAGMENTO 14 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 179 |
| IMAGEN 176. FRAGMENTO 14 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 180 |
| IMAGEN 177. DIRECTORIO TRAS EL FILTRADO DE MODO DE COLOR CIELAB. | 180 |
| IMAGEN 178. NOMBRE DE CADA IMAGEN TRAS EL FILTRADO DE MODO DE COLOR CIELAB. | 181 |
| IMAGEN 179. FRAGMENTO 16 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 182 |
| IMAGEN 180. DIRECTORIO DESPUÉS DE QUE LAS IMAGEN | 183 |
| IMAGEN 181. NOMBRE DE LA IMAGEN TRAS PASAR POR EL FILTRO SOBEL. | 184 |
| IMAGEN 182. FRAGMENTO 17 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 184 |
| IMAGEN 183. FRAGMENTO 18 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 185 |
| IMAGEN 184. FRAGMENTO 19 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 187 |
| IMAGEN 185. RED NEURONAL CONVOLUCIONAL. | 187 |
| IMAGEN 186. FUNCIÓN 'RELU LAYER'. | 188 |
| IMAGEN 187. MAX-POOLING. | 188 |
| IMAGEN 188. FRAGMENTO 20 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 189 |
| IMAGEN 189. FRAGMENTO 21 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 192 |
| IMAGEN 190. FRAGMENTO 22 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 193 |
| IMAGEN 191. FRAGMENTO 23 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 194 |
| IMAGEN 192. EXCEL DE LA MATRIZ DE CONFUSIÓN. | 195 |
| IMAGEN 193. FRAGMENTO 24 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 196 |
| IMAGEN 194. TABLA CON LOS PARÁMETROS CALCULADOS EN LAS LÍNEAS ANTERIORES. | 196 |

| | |
|--|-----|
| IMAGEN 195. FRAGMENTO 25 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 197 |
| IMAGEN 196. FRAGMENTO 66 DEL ALGORITMO IMPLEMENTADO EN MATLAB. | 197 |
| IMAGEN 197. CÁMARA HOSPITAL1-BARRANCOSECO. | 203 |
| IMAGEN 198. CÁMARA HOSPITAL2-CATEDRAL. | 204 |
| IMAGEN 199. CÁMARA HOSPITAL3-ISLETA. | 204 |
| IMAGEN 200. CÁMARA HOSPITAL4-ESCALERITAS. | 205 |
| IMAGEN 201. CÁMARA TAFIRA1-INFORMÁTICA. | 205 |
| IMAGEN 202. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 206 |
| IMAGEN 203. CÁMARA TAFIRA3-FOSCAM. | 206 |
| IMAGEN 204. CÁMARA TAFIRA4-FOSCAM. | 207 |
| IMAGEN 205. CÁMARA HOSPITAL1-BARRANCOSECO. | 207 |
| IMAGEN 206. CÁMARA HOSPITAL2-CATEDRAL. | 208 |
| IMAGEN 207. CÁMARA HOSPITAL3-ISLETA. | 208 |
| IMAGEN 208. CÁMARA HOSPITAL4-ESCALERITAS. | 209 |
| IMAGEN 209. CÁMARA TAFIRA1-INFORMÁTICA. | 209 |
| IMAGEN 210. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 210 |
| IMAGEN 211. CÁMARA TAFIRA3-FOSCAM. | 210 |
| IMAGEN 212. CÁMARA TAFIRA4-FOSCAM. | 211 |
| IMAGEN 213. CÁMARA HOSPITAL1-BARRANCOSECO. | 216 |
| IMAGEN 214. CÁMARA HOSPITAL2-CATEDRAL. | 216 |
| IMAGEN 215. CÁMARA HOSPITAL3-ISLETA. | 217 |
| IMAGEN 216. CÁMARA HOSPITAL4-ESCALERITAS. | 217 |
| IMAGEN 217. CÁMARA TAFIRA1-INFORMÁTICA. | 218 |
| IMAGEN 218. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 218 |
| IMAGEN 219. CÁMARA TAFIRA3-FOSCAM. | 219 |
| IMAGEN 220. CÁMARA TAFIRA4-FOSCAM. | 219 |
| IMAGEN 221. CÁMARA HOSPITAL1-BARRANCOSECO. | 220 |
| IMAGEN 222. CÁMARA HOSPITAL2-CATEDRAL. | 220 |
| IMAGEN 223. CÁMARA HOSPITAL3-ISLETA. | 221 |
| IMAGEN 224. CÁMARA HOSPITAL4-ESCALERITAS. | 221 |
| IMAGEN 225. CÁMARA TAFIRA1-INFORMÁTICA. | 222 |
| IMAGEN 226. CÁMARA TAFIRA2-MONTAÑAARUCAS. | 222 |
| IMAGEN 227. CÁMARA TAFIRA3-FOSCAM. | 223 |
| IMAGEN 228. CÁMARA TAFIRA4-FOSCAM. | 223 |
| IMAGEN 229. CÁMARA MODELO GRANDSTREAM GXV364_FHD. | 230 |
| IMAGEN 230. CÁMARA MODELO FOSCAM FI9901EP. | 232 |
| IMAGEN 231. PLUVIÓMETRO MODELO WXT530. | 234 |
| IMAGEN 232. PLUVIÓMETRO MODELO RG13. | 235 |

Índice de Tablas

| | |
|---|-----|
| TABLA 1. CLASIFICACIÓN DE LA ESTRUCTURA DE LLUVIA..... | 45 |
| TABLA 2. DATOS DE LA PRECIPITACIÓN DEL PLUVIÓMETRO MODELO VAISALA WXT530 | 51 |
| TABLA 3. DATOS DE LA PRECIPITACIÓN DEL PLUVIÓMETRO MODELO VAISALA WXT530 | 52 |
| TABLA 4. CARACTERÍSTICAS TÉCNICAS CÁMARAS GXV364_FHD..... | 59 |
| TABLA 5. CARACTERÍSTICAS TÉCNICAS CÁMARAS FOSCAM FI9901EP..... | 62 |
| TABLA 6. MATRIZ DE CONFUSIÓN BINARIA PARA LA DETECCIÓN DE LA PRECIPITACIÓN. | 110 |
| TABLA 7. MATRIZ DE CONFUSIÓN BINARIA CON LA INTERPRETACIÓN DE LOS RESULTADOS OBTENIDOS EN LA DETECCIÓN DE LA PRECIPITACIÓN. | 111 |
| TABLA 8. MATRIZ DE CONFUSIÓN PARA LA DETECCIÓN DE LA PRECIPITACIÓN..... | 113 |
| TABLA 9. MATRIZ DE CONFUSIÓN CON LA INTERPRETACIÓN DE LOS RESULTADOS OBTENIDOS EN LA DETECCIÓN DE LA PRECIPITACIÓN. | 113 |
| TABLA 10. MATRIZ DE CONFUSIÓN PARA LA DETECCIÓN DE LA PRECIPITACIÓN..... | 114 |
| TABLA 11. RESULTADO DE LA TABLA GAUSSIANA O TABLA DE LA NORMAL..... | 117 |
| TABLA 12. EJEMPLO DE PARÁMETRO DE LA CALIDAD DE LA CUANTIFICACIÓN CON SUS LÍMITES DE CONFIANZA Y SUS CORRESPONDIENTES VALORES CRÍTICOS [WITTEN 2005]..... | 118 |
| TABLA 13. RESULTADOS DE LA PRIMERA ITERACIÓN CON EL MODO DE COLOR LUMINANCIA EN TODAS LAS CÁMARAS. | 133 |
| TABLA 14. RESULTADOS DE LA PRIMERA ITERACIÓN CON EL MODO DE COLOR CIELAB EN TODAS LAS CÁMARAS..... | 138 |
| TABLA 15. INTERVALOS DE CONFIANZA DE LA MATRIZ BINARIA. | 148 |
| TABLA 16. MATRIZ BINARIA PARA LA DETECCIÓN DE LA PRECIPITACIÓN. | 194 |
| TABLA 17. MATRIZ DE CONFUSIÓN..... | 195 |
| TABLA 18. RESULTADOS DE LAS 9 ITERACIONES CON EL MODO DE COLOR LUMINANCIA. | 199 |
| TABLA 19. RESULTADOS DE LAS 9 ITERACIONES CON EL MODO DE COLOR CIELAB. . | 211 |
| TABLA 20. CARACTERÍSTICAS TÉCNICAS CÁMARAS GXV364_FHD..... | 230 |
| TABLA 21. CARACTERÍSTICAS TÉCNICAS CÁMARAS FOSCAM FI9901EP..... | 232 |
| TABLA 22. CARACTERÍSTICAS TÉCNICAS DEL PLUVIÓMETRO VAISALA WXT530. . | 235 |
| TABLA 23. CARACTERÍSTICAS TÉCNICAS DEL PLUVIÓMETRO | 236 |
| TABLA 24. PRECIOS Y COSTES DE AMORTIZACIÓN DE RECURSOS HARDWARE. | 245 |
| TABLA 25. PRECIOS Y COSTES DE AMORTIZACIÓN DE RECURSOS SOFTWARE. | 246 |
| TABLA 26. PRECIOS Y COSTES DE AMORTIZACIÓN DE RECURSOS SOFTWARE Y SOFTWARE TOTAL. | 247 |
| TABLA 27. PRECIOS Y COSTES DE AMORTIZACIÓN DE RECURSOS SOFTWARE Y SOFTWARE TOTAL Y TRABAJO TARIFADO POR TIEMPO EMPLEADO..... | 248 |
| TABLA 28. APLICACIÓN DE IMPUESTOS A LOS COSTES TOTALES. | 248 |

Índice de Ilustraciones

| | |
|---|-----|
| ILUSTRACIÓN 1. MAR DE NUBES , ISLAS CANARIAS..... | 46 |
| ILUSTRACIÓN 2. CLASIFICACIÓN DE LOS PLUVIÓGRAFOS ANALÓGICOS. | 49 |
| ILUSTRACIÓN 3. POSIBLES REGIONES DE INTERÉS EN LAS QUE HEMOS DIVIDIDO LA IMAGEN..... | 100 |

Acrónimos

| | |
|-------|---|
| AEMET | Agencia Estatal de Meteorología. |
| ITU | Organización Internacional de las Telecomunicaciones. |
| TFM | Trabajo fin de máster. |
| ROIs | Regions of Interest. |
| OMM | Organización Meteorológica Mundial. |
| LAB | Componentes de color L, A y B. |
| CNN | Red Neuronal Convolutiva. |

Parte I

Memoria

Capítulo 1:

1. Introducción

En el este capítulo se hará una breve descripción de los sistemas de detección y cuantificación de lluvia disponibles actualmente. Posteriormente, se estudiará una alternativa a los sistemas de observación ya existentes para aportar una ampliación de la cobertura espacial actual y un muestreo temporal mediante las imágenes captadas desde una red de cámaras de videovigilancia. También se describirán los objetivos del Trabajo Fin de Máster y la estructura de la memoria.

1.1 Antecedentes del trabajo

Desde siempre los seres humanos han mostrado un gran interés en la meteorología para multitud de fines, como son el cuidado de los cultivos, el transporte, tanto aéreo como marítimo, o la protección de las personas ante desastres naturales, entre otros muchos casos. Por tanto, observar y predecir las condiciones meteorológicas es de importancia vital.



Imagen 1. Sistemas de Observación Meteorológicas [OMM 2019].

Poco a poco, mediante la incorporación de nuevas tecnologías, se realizan mejores estimaciones y predicciones del clima. Actualmente, la red de observación de la Organización Meteorológica Mundial (OMM) desarrolla proyectos para mejorar los sistemas de observación actuales. En los últimos años, OMM ha incorporado nuevas tecnologías (mostradas en la imagen 1), tales como, estaciones meteorológicas, radares meteorológicos, satélites meteorológicos, boyas meteorológicas, los buques de observación marítima, las radiosondas, entre otros.

El uso de estos sistemas conlleva un gran costo para su instalación, puesta en funcionamiento y mantenimiento. En este sentido, una de las ideas que subyace en este proyecto es la de avanzar hacia una ampliación de los sistemas de observación con un coste lo más bajo posible. Trabajaremos sobre la idea de dar valor añadido a sistemas de observación ya existentes. Es el caso de los sistemas de videovigilancia, basados en observación con cámaras IP, etc.

Hoy en día los sistemas de videovigilancia son parte del ecosistema tanto de las ciudades como en zonas rurales, espacios naturales, etc. Normalmente, son utilizados con el objetivo principal de seguridad de bienes y personas, control de tráfico, turístico, etc. Por otro lado, desde hace décadas los aficionados a la meteorología incluyen pequeñas cámaras web en sus estaciones meteorológicas. Podría decirse que es aquí donde comienza el uso generalizado de los sistemas de videovigilancia en aplicaciones meteorológicas. Con todo, el espectro de cámaras utilizables para vigilancia meteorológica es amplio.

Dos de las grandes ventajas en el uso de esta tecnología son:

- Abaratar los costes que implica los sistemas meteorológicos de uso específico.
- Disponer de un gran despliegue de videocámaras en amplias zonas ya que muchos sistemas de videovigilancia pueden ser cedidos por organismos oficiales, empresas privadas o aficionados a la meteorología a través de la red. No obstante, si fuera necesario, podrían instalarse cámaras con el objetivo específico de hacer observación meteorológica.

En la actualidad, el uso de las cámaras (por ejemplo, webcams) en el ámbito de la meteorología se basa en el análisis de la atmósfera para conocer su estado en cualquier momento mediante la observación y la toma de datos meteorológicos (magnitudes físicas).

Una de las formas de observación actualmente en auge es mediante algoritmos para la detección de lluvia mediante imágenes. Un ejemplo, aplicado a la eliminación del efecto de la lluvia en imágenes, es el realizado en la Universidad de Columbia (USA) y en la Universidad de Xiamen (China) [Xueyang 2017]. Su trabajo se basa en la eliminación de la lluvia en imágenes individuales captadas desde cámaras basándose en la observación de las pérdidas de detalles en alta frecuencia de la imagen. Cuando una imagen contiene lluvia puede presentar una pérdida muy significativa de altas frecuencias. Todo este proceso se integra en un sistema de aprendizaje mediante el uso de redes neuronales convolucionales. Su resultado se muestra en la imagen 2.



Imagen 2. Resultado del algoritmo detección de lluvia [Xueyang 2017].

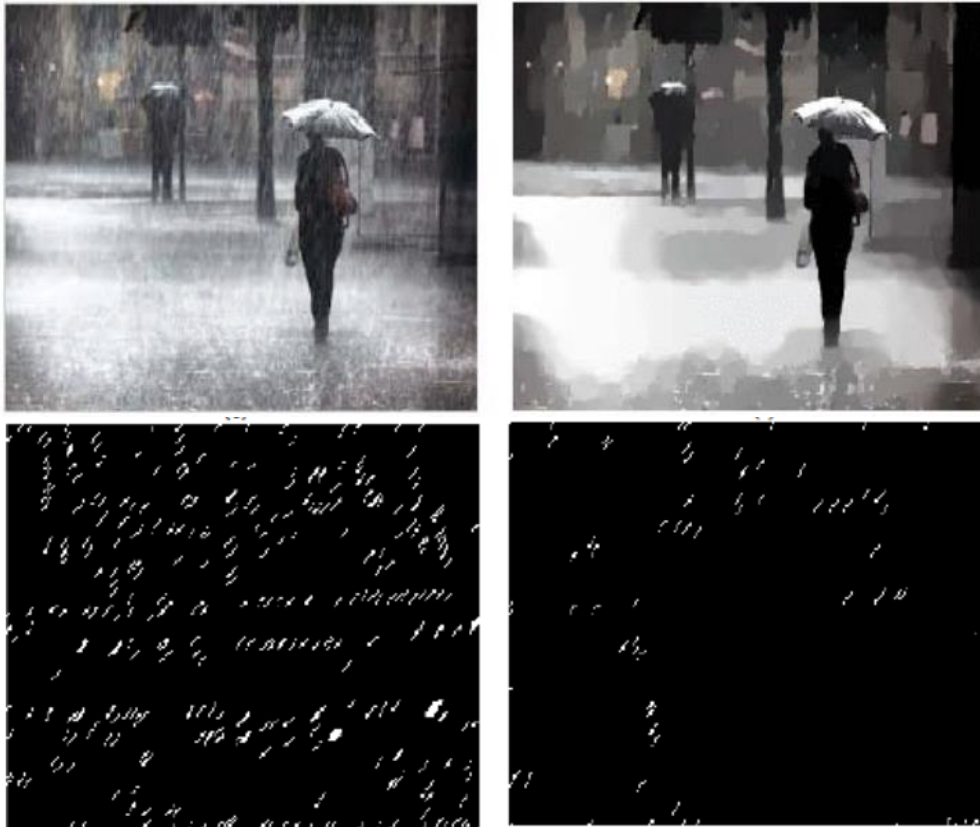


Imagen 3. Técnica de procesado de señal y minimización de gradientes [Manu 2015].

Otro ejemplo es el presentado en Bangalore (India) [Manu 2015], donde se ha realizado un estudio similar al anterior pero basado en el uso de técnicas de procesado de señal y minimización de gradientes. Como se observa en la imagen 3 se muestra dos ilustraciones con sus gradientes correspondientes. La ilustración de la izquierda es la imagen original con lluvia y a la derecha la imagen que se le ha aplicado el algoritmo de gradientes para la eliminación de lluvia.

El proceso de detección está abierto a realizar un tratamiento de estabilización y rectificación de imágenes debido a la eventual acción del viento sobre las cámaras y la variabilidad de las condiciones de iluminación. Por ejemplo, en la Universidad de Washington, Seattle, en el laboratorio de física aplicada, los físicos Michael Schwendeman y Jim Thomson [Schwendeman 2015] realizaron un estudio sobre los problemas de estabilización de los videos digitales en plataformas flotantes. Elaboraron un método basado en una técnica de seguimiento de la línea del horizonte que se probó bajo una variedad de condiciones de iluminación y estado del mar. Como se observa en la imagen 4, mediante la línea del horizonte se calcula la incidencia y ángulos de rectificación de las cámaras ubicadas en la plataforma flotante sin la intervención de estaciones terrenales u otros sistemas externos.

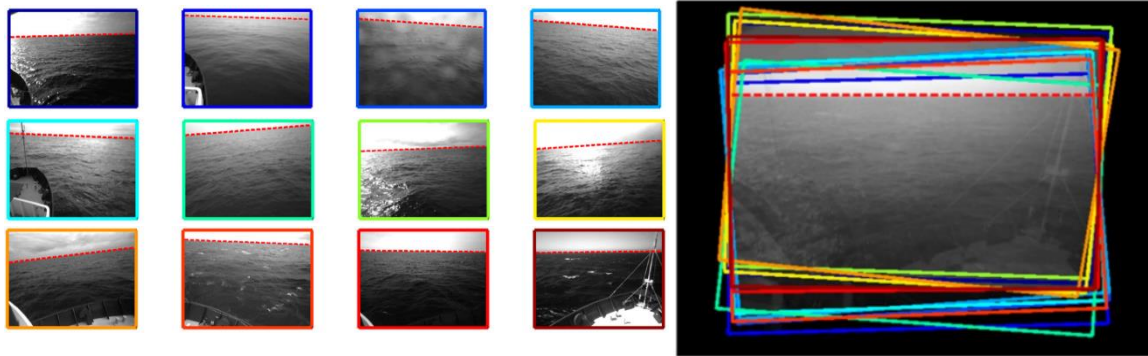


Imagen 4. Técnica de seguimiento de la línea del horizonte probada bajo una variedad de condiciones de iluminación y estado del mar. En la última imagen se representa la estabilización de las diferentes imágenes captadas [Schwendeman 2015].

Este TFM es un proyecto que se basa en sistemas de videovigilancia meteorológica para el seguimiento de riesgos medioambientales. Va más allá de planteamientos como los expuestos anteriormente, y proponemos un funcionamiento alternativo y complementario a los métodos tradicionales de observación de tal forma que se aporta nuevo valor a la red de vigilancia.

Las líneas de trabajo que se desarrollarán están integradas en el Proyecto VIMETRI-MAC [VIMETRI-MAC], y es de potencial uso con cámaras de videovigilancia, propias y también públicas o privadas, desplegadas por las Islas Canarias para la detección de eventos de lluvia. Con este sistema se obtienen una serie de imágenes que se utilizarán para reconocimiento de patrones e identificar dichos eventos en dos fases: Detección y posterior cuantificación.

En términos generales, el valor añadido y novedoso que se aporta en este proyecto es ampliar la cobertura espacial y el muestreo temporal de un sistema de predicción meteorológica de lluvia mediante las imágenes captadas desde una red de cámaras de las estaciones de videovigilancia.

Como parte del núcleo principal se desarrollarán técnicas de procesamiento de señal y se hará uso de técnicas de Aprendizaje Máquina (Machine Learning y Deep Learning) [Sebe 2005] [Soo-Kyun 2015] [Yao 1991]. Así mismo, partimos de la tesis doctoral de Iván Guerra Moreno [Guerra 2017].

1.2 Objetivos

Objetivo principal:

Implementación de algoritmos de Machine Learning y Deep Learning para la detección y cuantificación de precipitación de lluvia basado en imágenes de cámaras distribuidas por las zonas de observación.

Aprendizaje Profundo (en inglés, Deep Learning) es un conjunto de algoritmos de Aprendizaje Automático (en inglés, Machine Learning) que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial.

Objetivos secundarios:

Para alcanzar el citado objetivo es necesario identificar otros más específicos que, vinculados a este permitan alcanzarlo. Así, distinguimos los siguientes:

1. Estudio de las características necesarias de los dispositivos de captación de imágenes.

A efectos de alcanzar una adecuada calidad de imagen para el procesado es necesario realizar un estudio previo de las distintas cámaras IPs y webcams disponibles en el mercado. Haciendo reseña de sus pros y sus contras a la hora de realizar seguimiento meteorológico. Entre estas características podemos destacar los de resolución en píxeles y tasas de transferencia de imágenes.

2. Estudio de los distintos modos de almacenamiento y codificaciones tanto de imagen como de video.

Una vez determinadas las características de las cámaras e imágenes necesarias nos plantearemos las distintas técnicas de almacenamiento de las mismas y sus distintas codificaciones. Existen multitud de técnicas distintas de almacenamiento, ya sean frame a frame cada cierto tiempo, video continuo o video discontinuo. Así mismo los archivos generados utilizaran tanto un formato de almacenamiento (mp4, mpeg, avi, jpg, gif, png, etc) como una codificación (Xvid, Theora, RealVideo, WMV, MPEG-1, MPEG-2, MPEG-4, H264, entre otros) [Soo-Kyun 2015]. Estas elecciones nos permitirán realizar el cálculo de los medios necesario para almacenamiento y la cantidad posible de datos a almacenar.

Estudio de los procesos previos a la aplicación de las imágenes. Por ejemplo, preprocesado mediante balance de blancos [Yao 1991], reducción, estabilización [Skow 2006], etc.

Una vez tenemos determinado que tipo de imágenes vamos a tratar deberemos realizar un preprocesado de normalización de las distintas imágenes dado que las características de luz incidente en cada localización varía con las horas del día y durante los días del año. Por otro lado, se producen variaciones en la imagen a la hora de seleccionar las regiones de interés (ROIs) debido a efectos como los vientos o vibraciones que hacen que las distintas imágenes se encuentren desestabilizadas.

3. Estudio de los algoritmos de reconocimiento de imágenes y detección de eventos. [Aguilar 2014] [Thomson 2009].

1.3 Estructura de la memoria

Este proyecto fin de máster consta de tres partes fundamentales:

- Memoria
- Pliego de condiciones
- Presupuesto

La memoria contiene los siguientes capítulos:

Capítulo 1: Introducción. En este capítulo se hace una breve descripción sobre los sistemas de detección y cuantificación de lluvia disponibles actualmente. Posteriormente se estudia una alternativa a los sistemas de observación ya existentes para aportar una ampliación de la cobertura espacial actual y un muestreo temporal mediante las imágenes captadas desde una red de cámaras de las estaciones de videovigilancia. También se describen los objetivos del proyecto fin de máster y la estructura de la memoria.

Capítulo 2: La precipitación. Se realizará un estudio sobre la precipitación: Su definición, su formación, su clasificación según su naturaleza física, mecanismos de formación, formación de meteoros y estructura, medidores actuales y análisis de las variables necesarias para la cuantificación de la lluvia.

Capítulo 3: La base de datos. Se detalla la creación y organización de la base de datos utilizada para la detección y la cuantificación de la precipitación de la lluvia.

Capítulo 4: Procesado y extracción de características de la imagen. Se estudia la etapa de preprocesado previa por la que pasaremos las imágenes antes de ser entregadas a la red neuronal convolucional. Este preprocesado consta de varios bloques, algunos de ellos son el balance de

blancos, extracción de regiones de interés, filtros para cambios de modo de color (luminancia y Cielab) y filtrado de Sobel.

Capítulo 5: Marco metodológico para la clasificación de las imágenes.

Se estudiará el problema complejo de la monitorización meteorológica. Para ello, emplearemos el aprendizaje máquina para la detección y cuantificación de la precipitación de la lluvia basándonos en imágenes recibidas desde las cámaras de vigilancia.

Capítulo 6: Método de detección y cuantificación de precipitación a partir de imágenes.

Se expone el método que hemos utilizado para detectar y cuantificar el nivel de precipitación de lluvia a partir de imágenes recibidas desde diferentes cámaras de videovigilancia. Particularmente, para los experimentos utilizamos cámaras ubicadas en varios puntos de la isla de Gran Canaria. Mostraremos paso a paso todos los elementos constituyentes desde el preprocesado hasta la obtención de resultados, pasando por la Red Neuronal Convolutiva empleada, el diseño de los experimentos y los resultados obtenidos.

Capítulo 7: Conclusiones y líneas futuras.

Se exponen, por un lado, las conclusiones obtenidas en este TFM. Mostramos los objetivos planteados inicialmente y los resultados obtenidos de los diferentes experimentos realizados. Y, por otro lado, se detallan propuestas de líneas de investigación futuras relacionadas con la implementación y desarrollos de sistemas de aprendizaje máquina para la detección y cuantificación de la lluvia sobre imágenes de videovigilancia.

Capítulo 2:

2. La precipitación

En este capítulo presentamos un estudio sobre la precipitación. Particularmente, abordamos su definición, formación, clasificación según su naturaleza física, mecanismos de formación, formación de meteoros y estructura, medidores actuales y análisis de las variables necesarias para la cuantificación de la lluvia.

2.1 Definición de precipitación

Para definir correctamente la precipitación es necesario definir previamente meteoro e hidrometeoro, además de otros fenómenos asociados. A tal efecto recogemos aquí lo que La Organización Meteorológica Mundial (OMM) define el meteoro de la siguiente forma [OMM 2019]:

“Un meteoro es un fenómeno observado en la atmósfera o sobre la superficie de la tierra, que consiste en una suspensión, una precipitación, o un depósito de partículas líquidas, acuosas o no, o de partículas sólidas, o un fenómeno de la naturaleza de manifestación óptica o eléctrica”.

Dependiendo de la naturaleza de las partículas que constituye los meteoros y de los procesos físicos que implica su ocurrencia, se pueden clasificar en cuatro grupos: hidrometeoros, litometeoros, fotometeoros y electrometeoros [OMM 2019].

- **Hidrometeoro:**

“Un hidrometeoro es un meteoro que consiste en un conjunto de partículas de agua líquida o sólida, suspendidas en la atmósfera o cayendo a través de ella, o que son empujadas por el viento desde la superficie de la tierra, o depositadas sobre objetos que se encuentran sobre el suelo o en el aire libre.” [OMM 2019]

Por ejemplo: lluvia, nieve y granizo.

- **Litometeoros:**

“Un litometeoro es un meteoro que consiste en un conjunto de partículas, la mayoría de las cuales son sólidas y no acuosas. Las partículas están más o menos suspendidas en el aire, o son levantadas por el viento desde el suelo.” [OMM 2019]

Por ejemplo: la calima de polvo, tormenta de arena y el humo.

- **Fotometeoros:**

“Un fotometeoro es un fenómeno luminoso producido por la reflexión, refracción, difracción o interferencia de luz proveniente del Sol o la Luna.” [OMM 2019]

Por ejemplo: arco iris, espejismos y rayos crepusculares.

- **Electrometeoros:**

“Un electrometeoro es una manifestación visible o audible de la electricidad atmosférica. Los electrometeoros corresponden a descargas eléctricas discontinuas (relámpagos, truenos) o se producen como fenómenos más o menos continuos (fuego de San Telmo, aurora polar).” [OMM 2019]

Por ejemplo: relámpagos, truenos, fuego de San Telmo y aurora polar.

Por lo tanto, la precipitación consiste en la caída desde la atmósfera hasta la superficie de la tierra o evaporarse completamente mientras caen un conjunto de partículas denominadas anteriormente como hidrometeoros, ya sea en estado líquido, tales como la lluvia o llovizna, o en estado sólido tal como la nieve, cinarra, nieve granulada, polvo diamante, hielo granulado y granizo.

2.2 Formación de la precipitación

La formación de la precipitación sigue un ciclo denominado ciclo hidrológico o ciclo del agua. Este ciclo se basa en un proceso de circulación del agua entre las diferentes capas de la hidrosfera donde se minimiza las reacciones químicas, ya que sólo se efectúa el traslado del agua con la ayuda del cambio de su estado físico. Estos ciclos mantienen una cantidad de agua constante pero su distribución no es uniforme, va cambiando. [Sociedad Geográfica de Lima 2011] [Wikipedia 2020].

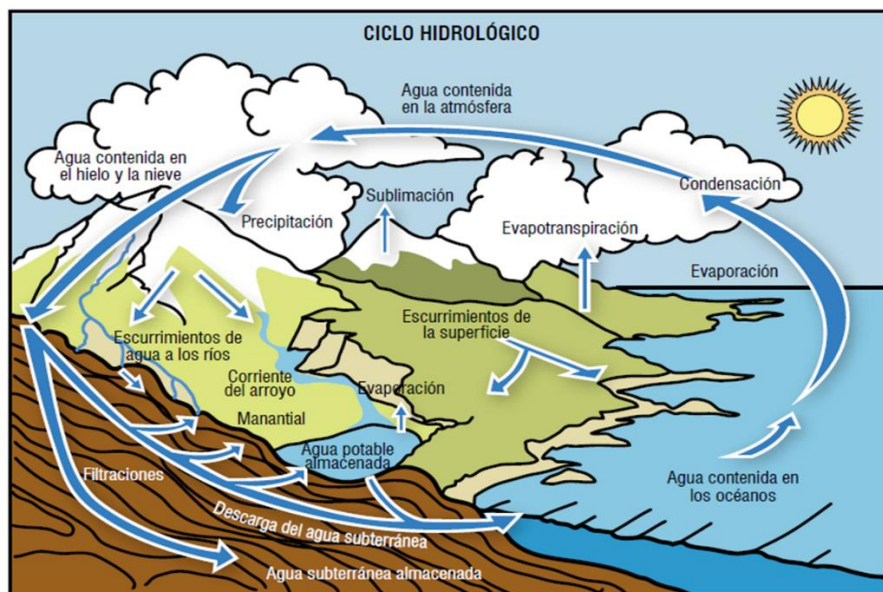


Imagen 5. Ciclo hidrológico [Sociedad Geográfica de Lima 2011].

Como se ilustra en la imagen 5, los pasos que desarrolla el ciclo hidrológico son los siguientes: [AEMET 2020] [Wikipedia 2020].

- **Evaporización:** El agua presente en la superficie terrestre se evapora gracias al calor del sol del que toma su energía. Esto da lugar a un cambio del estado físico del agua, pasar de un estado líquido a uno sólido.

Otro de los factores que contribuyen a evaporización del agua en un 10% es la sudoración de los animales y la transpiración de las plantas.

- **Condensación:** El vapor de agua va ascendiendo entre las diferentes capas de la hidrosfera, al ir ascendiendo se va enfriando y esto ocasiona que el agua se vaya condensando, es decir, cambia su estado físico de un estado de vapor a un estado líquido. Genera diversas moléculas de agua denominadas aerosoles de diferentes tamaños (entre 0.1 a 10 μ m de

diámetro) que se unen formando núcleos de condensación. Esta unión da lugar a la formación de nubes, niebla y cristales de hielo.

- **Precipitación:** a medida que van ascendiendo los aerosoles van incrementando su tamaño por condensación hasta un punto en el que son lo suficientemente pesados como para precipitarse, esto ocurre cuando tienen un diámetro superior a 0.1mm. Parte de estas gotas en el descenso decrecen debido a la baja velocidad de su caída y vuelven a evaporizarse antes de alcanzar la superficie de la tierra y otras incrementan su tamaño debido al impacto y agregación y, además, poseen una velocidad mayor, es decir, superior a los 3 m/s, dando lugar a las gotas de lluvia.
- **Infiltración:** Las gotas de lluvia que llegan a la superficie de la tierra penetra el suelo y se une a aguas subterráneas.
- **Fusión:** Proceso a través del cual el agua en estado sólido que ha alcanzado las zonas altas de la superficie terrestre, tal como el granizo o la nieve, cambia de estado sólido a líquido, es decir, se produce el fenómeno del deshielo. De esta forma se encausa el agua por diferentes ríos hasta alcanzar el mar.
- **Escorrentía:** Debido a la fuerza de la gravedad las moléculas de agua se transportan desde las zonas altas de la superficie terrestre hasta los mares y océanos. Formando así los ríos y caudales subterráneos que originan el arrastre de sedimentos y provocan erosiones en la superficie terrestre. A partir de esta fase comienza el ciclo del agua nuevamente.

2.3 Clasificación de la precipitación

Existen diferentes formas de clasificar la precipitación, en este TFM nos centraremos en cuatro clasificaciones principales. Estas son según:

- Su naturaleza física
- El mecanismo de formación
- La formación del meteoro o gota
- La estructura

2.3.1 Clasificación según las características físicas

Esta clasificación se centra en una descripción de los meteoros que forman la precipitación. Los cuales, pueden adquirir diferentes formas y tamaños dependiendo del proceso de condensación recibido. Los más comunes son los siguientes: [OMM 2019]

- **Lluvia:** precipitación de las gotas de agua en estado líquido que poseen entre 0.1 y 5 mm de diámetro. Según su intensidad y duración se puede distinguir cuatro grandes grupos:
 - **Llovizna:** el diámetro de las gotas de agua es inferior a 0.5mm. tienen poca intensidad y poca duración.
 - **Chubascos:** precipitación de gran intensidad y corta duración.
 - **Aguacero:** precipitación de poca intensidad y larga duración.
 - **Trombas:** precipitación de gran intensidad y gran duración.

- **Nieve:** precipitación de las gotas de agua en estado sólido con formas geométricas particulares. Según su diámetro se puede distinguir cuatro grandes grupos:
 - **Nieve:** precipitación en forma de cristales de hielo. Su forma es hexagonal. Su tamaño puede alcanzar centímetros de diámetro.
 - **Escarcha:** gránulos de hielo de gotas de agua que se han congelado sobre objetos de la superficie terrestre.
 - **Granizo:** precipitación en forma de bolas de hielo. Según su forma pueden ser esferoidales, cónicos o irregulares. Su tamaño está comprendido entre 5-125mm de diámetro.

2.3.2 Clasificación según el mecanismo de formación

Esta clasificación tiene en cuenta la masa de aire que genera la precipitación y el grado de humedad que contiene la atmósfera. Se distinguen tres tipos de precipitación según su mecanismo de formación [OMM 2019].

- **Precipitación convectiva:**
Como se muestra en la imagen 6, los rayos solares inciden sobre la superficie terrestre y aumentan su temperatura. El aire próximo a la superficie aumenta su temperatura y se vuelve más ligero, generando vapor de agua. Esto ocasiona que dicho aire ascienda a capas superiores de la atmósfera originando la formación de nubes de desarrollo vertical tipo cúmulos y cumulonimbos (imagen 7). El vapor de agua que se encuentra a mayor altura se enfría y da lugar a fuertes precipitaciones de

lluvia o ligeras de corta duración, de tipo chubascos. También pueden generar granizos y tormentas.

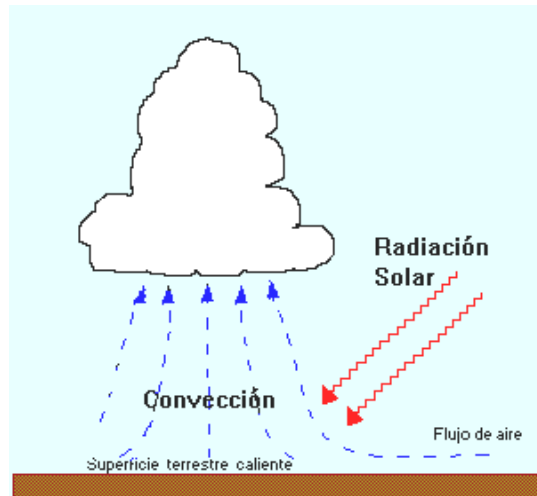


Imagen 6. Precipitación convectiva.

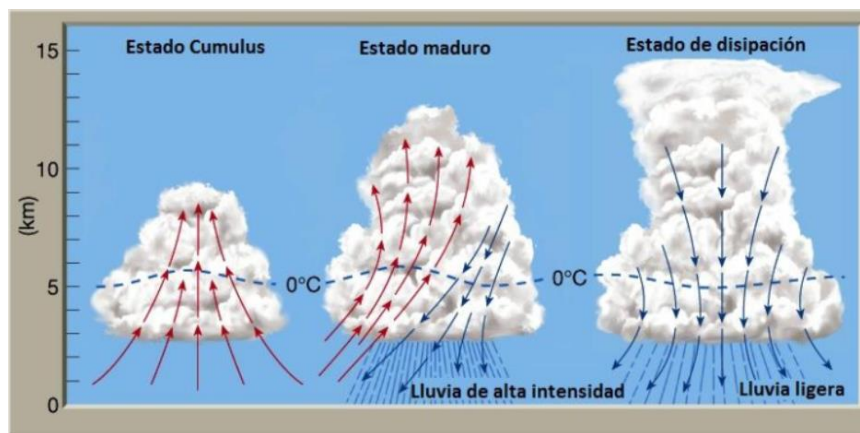


Imagen 7. Formación de nubes en precipitaciones convectivas.

- **Precipitación orográfica:**

Como se ilustra en la imagen 8, este tipo de precipitación tiene lugar cuando corrientes oceánicas de aire se encuentran con un obstáculo, por ejemplo, con barreras montañosas. Este obstáculo obliga a estas corrientes con masa de aire húmedo a ascender generando una elevación mecánica del aire. Este ascenso por el obstáculo provoca que el aire se enfríe y, debido a esto, el vapor de agua se condensa y se precipita. En el descenso de la otra ladera del obstáculo se calienta la masa de aire y origina un tiempo seco y cálido.

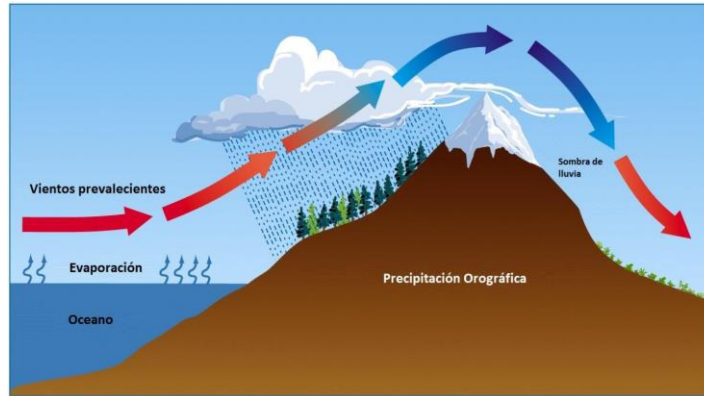


Imagen 8. Precipitación orográfica.

- **Precipitaciones frontales:**

Este tipo de precipitaciones tiene lugar cuando se producen dos masas de aire, de características diferentes (masa de aire cálido y masa de aire frío), y se ponen en contacto. La zona de contacto se le denomina frente. Existen dos tipos de frentes:

- **Frente frío o precipitaciones ciclónicas:**

Como se ilustra en la imagen 9, en el frente frío el desplazamiento de ambas masas de aires están diferentes sentidos, es decir, sentido contrario. La masa de aire frío eleva la masa de aire caliente creando diferencias de presiones y generando nubes y lluvias intensas y chubascos tormentosos.

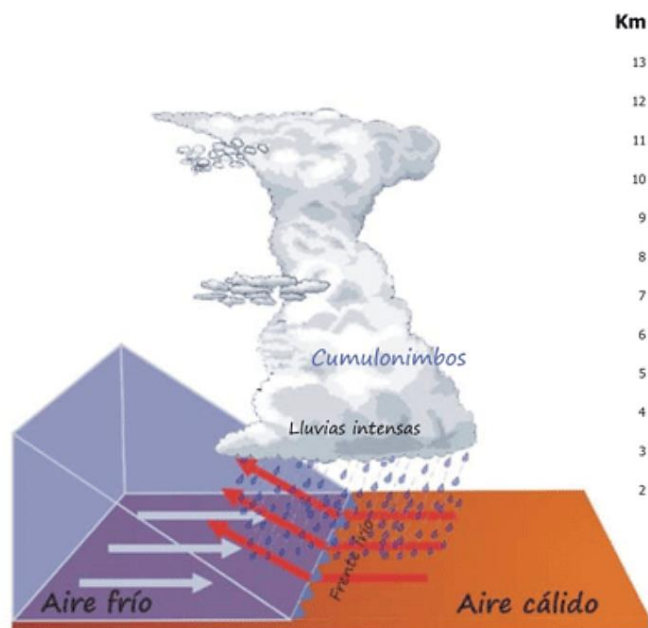


Imagen 9. Frente frío.

- **Frente cálido o precipitaciones anticiclónicas:**

Como se ilustra en la imagen 10, en el frente cálido el desplazamiento de ambas masas de aires tiene el mismo sentido.

En este tipo de frente la masa de aire cálido es menos densa, y por lo tanto, más ligera que la masa de aire frío. Por ello, la masa de aire cálido asciende sobre la masa de aire frío, al ascender la masa de aire cálido se enfría y se condensa. Esto ocasiona lloviznas y lluvias moderadas.

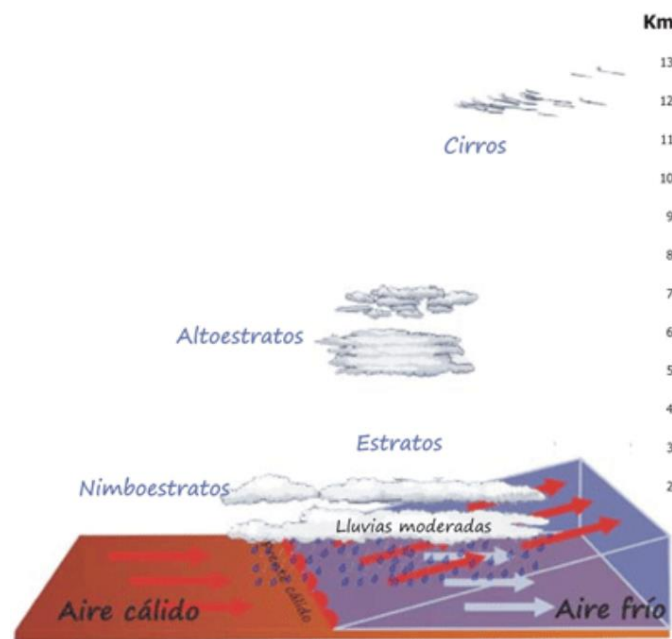


Imagen 10. Frente cálido.

2.3.3 Clasificación según la formación del meteoro

La precipitación está formada por un conjunto de gotas de agua en estado líquido. Para un mejor estudio de la precipitación, debemos conocer con más detalle cómo clasificar las gotas de agua según su velocidad, forma u orientación, tamaño y la temperatura [González 2017].

- **Velocidad:**

En 1949, Gunn y Kinzer [Gunn 1949] realizaron un modelo matemático para el cálculo de la velocidad de la gota de la precipitación. Llevaron a la conclusión que el radio de la gota estaba relacionado con su velocidad, de hecho, tras varias investigaciones observaron que la velocidad de la gota aumenta cuanto mayor es su radio. Este modelo se ilustra en la imagen 11, donde se observa la relación entre el radio de la gota y su velocidad. Un detalle importante que se observa en este modelo es que

si el radio supera, aproximadamente, los 2,5mm la gota no sigue aumentando su velocidad. Es decir, la velocidad máxima de la gota es de 9m/s.

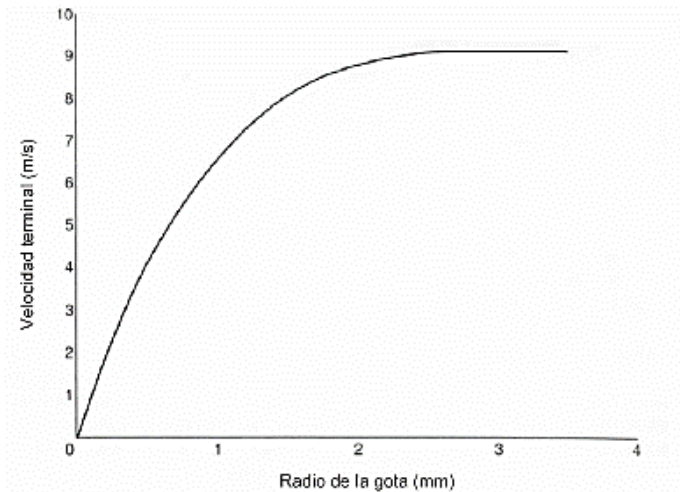


Imagen 11. Modelo de Gunn y Kinzer.

A demás, según AEMET [AEMET 2018] para que la gota de lluvia llegue hasta la superficie terrestre debe de tener como mínimo un diámetro de 0,5mm, es decir, un radio mínimo de 0,25mm. Si su radio es menor al indicado la lluvia no alcanza la superficie terrestre y recibe el nombre de virga (imagen 12).

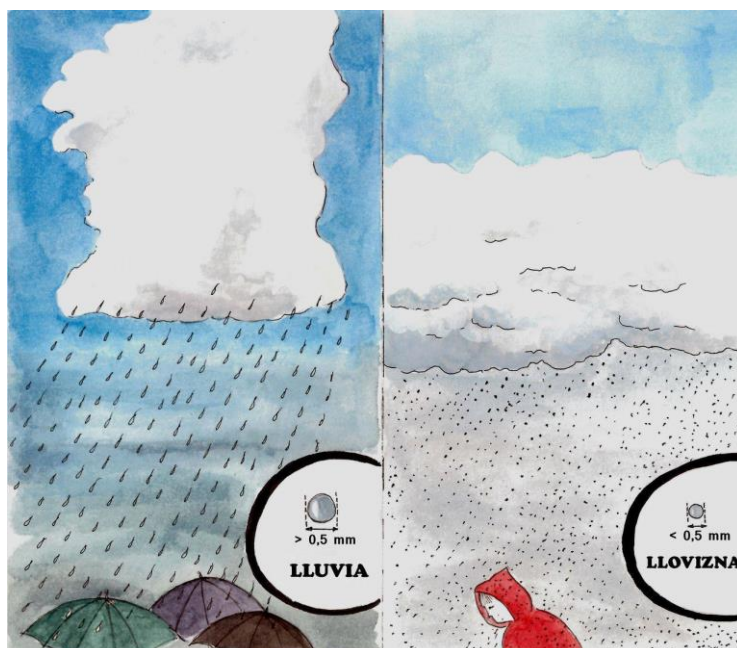


Imagen 12. Diámetro de la gota de lluvia.

- **Temperatura:**

La temperatura de la gota de la precipitación depende de la temperatura de la capa atmosférica a la que se encuentra sometida externamente. Aproximadamente, su temperatura oscila desde por debajo de los 0°C hasta 27°C [AEMET 2018].

- **Forma u orientación:**

Las gotas de agua están sometidas a la fuerza de la gravedad, así como también, a la tensión superficial del agua. Estos dos efectos ocasionan que posea una forma esferoidal. Para caracterizar correctamente su forma y orientación es necesario introducir dos modelos:

- **Modelo de Pruppacher y Pitter** [Pruppacher 1971]: Este modelo se basa en el estudio de la forma de la gota de la precipitación y relacionarla con su tamaño. Tras realizar varios experimentos observaron que el tamaño de la gota está relacionado con la forma de la misma. El tamaño de la gota está comprendido entre 100µm y 3.5mm de radio. En la imagen 13 se muestra los resultados de este modelo:

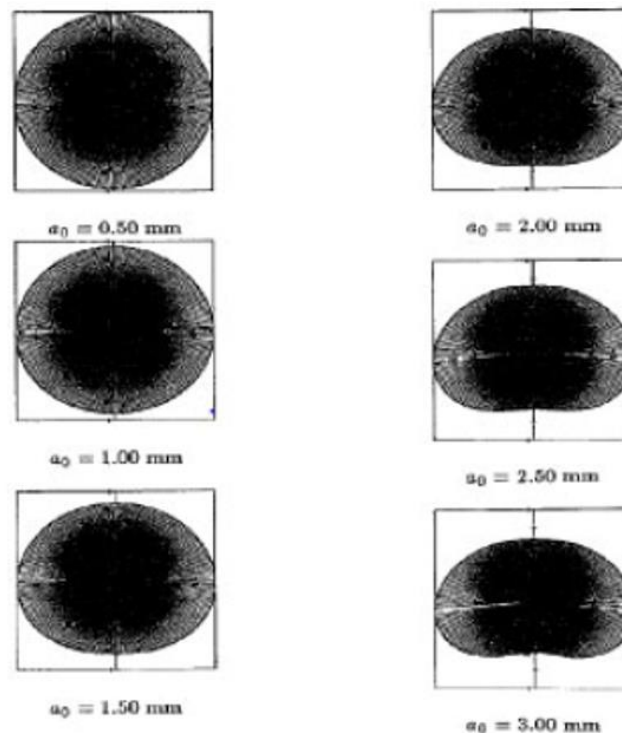


Imagen 13. Forma de la gota de la precipitación.

- **Modelo de G. Brussaard y P. A. Watson** [Brussaard 1995]: Este modelo incluyó el efecto de la fuerza gravitatoria y el efecto aerodinámico a la que está sometida la gota para estudiar la orientación de la gota. La conclusión que se obtuvo es que el

ángulo de inclinación de la gota de la precipitación disminuye con la altura. Figura 14:

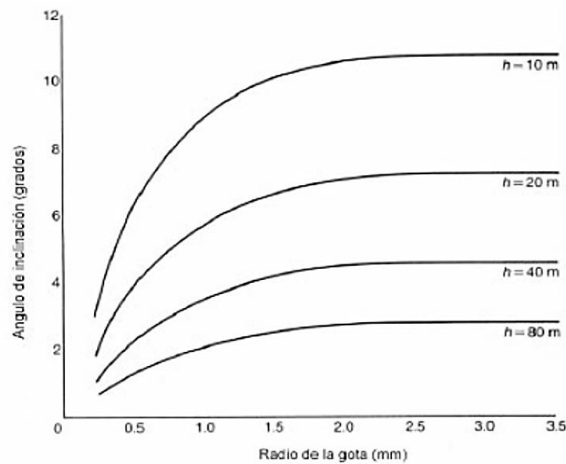


Imagen 14. Ángulo de inclinación según el radio de la gota de la precipitación.

2.3.4 Clasificación según la estructura

La precipitación se puede clasificar según su estructura vertical u horizontal, para un mejor estudio de espacio alcanzado y el tiempo de duración de la misma. La clasificación se puede observar en la siguiente tabla 1 [González 2017].

Tabla 1. Clasificación de la estructura de lluvia

| | Estructura de lluvia | |
|--|--------------------------|---|
| | Horizontal | Vertical |
| Región de distribución de la precipitación | Amplia | Reducida, pero se extiende desde la superficie terrestre hasta una altura de la isoterma de 0°C |
| mecanismo de formación | Precipitación orográfica | Precipitación convectiva y frontales |
| Cantidad de precipitación | Ligera | Intensa |
| Homogeneidad | No homogénea | Si homogénea |
| Tipo de precipitación | Débil | Fuerte |

Las Islas Canarias están ubicadas en una región considerada subtropical. Las islas reciben los vientos alisios, un viento húmedo procedente del noreste. Debido a los vientos alisios a el que están expuestas, se obtiene una capa superficial húmeda superpuesta a una capa seca, generando un cambio brusco de temperatura. Este cambio da lugar a nubes de tipo estratocúmulos, e impide el desarrollo de nubes de desarrollo vertical o nubes de grandes cúmulos. Este tipo de nube es conocida en Canarias como “el mar de nubes” (Ilustración 1). Esto ocasiona escasas precipitaciones.

Sin embargo, las islas cuentan con una gran variedad de vegetación: Unos de los más destacados “los bosques de laurisilva”. La captación del agua que obtiene esta vegetación se debe a las precipitaciones de estructura de lluvia horizontal. Da lugar a una presencia de forma continuada de nubes bajas que al encontrar obstáculos (zonas montañosas) obliga a estas nubes a ascender generando una elevación mecánica del aire. Este ascenso por el obstáculo provoca que el aire se enfríe y, debido a esto, el vapor de agua se condensa y se precipita sobre la vegetación o por el contacto de las nubes con la vegetación éstos se empapan de agua de precipitación.

Un ejemplo son Las Breñas en La Palma, se registra entre 1000 y 1500 litros de agua al año [AEMET 2021].



Ilustración 1. Mar de nubes , Islas Canarias.

2.4 Medidores de la precipitación actuales

Actualmente, para cuantificar las precipitaciones es necesario el uso de determinados dispositivos o sistemas, tales como, pluviómetros, pluviógrafos, redes pluviométricas y estaciones climatológicas. A continuación, se realizará

una breve descripción de cada uno de estos dispositivos o sistemas de captación y cuantificación de la precipitación [USAL 2020] [Velázquez 2014]:

- **Pluviómetro:**

El pluviómetro es un recipiente cilíndrico graduado en milímetros (mm) o, lo que es lo mismo, en litros por metro cuadrado (l/m^2) que recoge el agua de la precipitación. El tamaño del recipiente influye de manera poco significativa en la medida de la cuantificación de la precipitación, pero su ubicación es de vital importancia para la medición ya que ésta si influye en mayor grado en las medidas resultantes obtenidas.

En España, la norma establece que la boca del pluviómetro debe tener 200cm^2 y, en cuanto a su ubicación, indica una altura sobre el suelo de 1,5m. Este dispositivo debe estar alejado de zonas de arboledas o construcciones elevadas, como mínimo el doble de la altura del obstáculo. Se debe de elegir un lugar que no esté expuesto a un viento excesivo. Como se ilustra en la imagen 15.

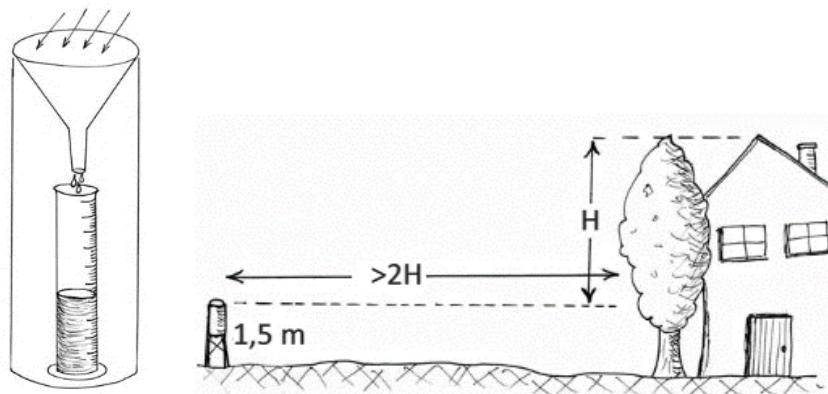


Imagen 15. En la izquierda, pluviómetro con embudo para aumentar el área de captación del agua de la precipitación. En la derecha, ubicación del pluviómetro establecida por la norma en España. [USAL 2020]

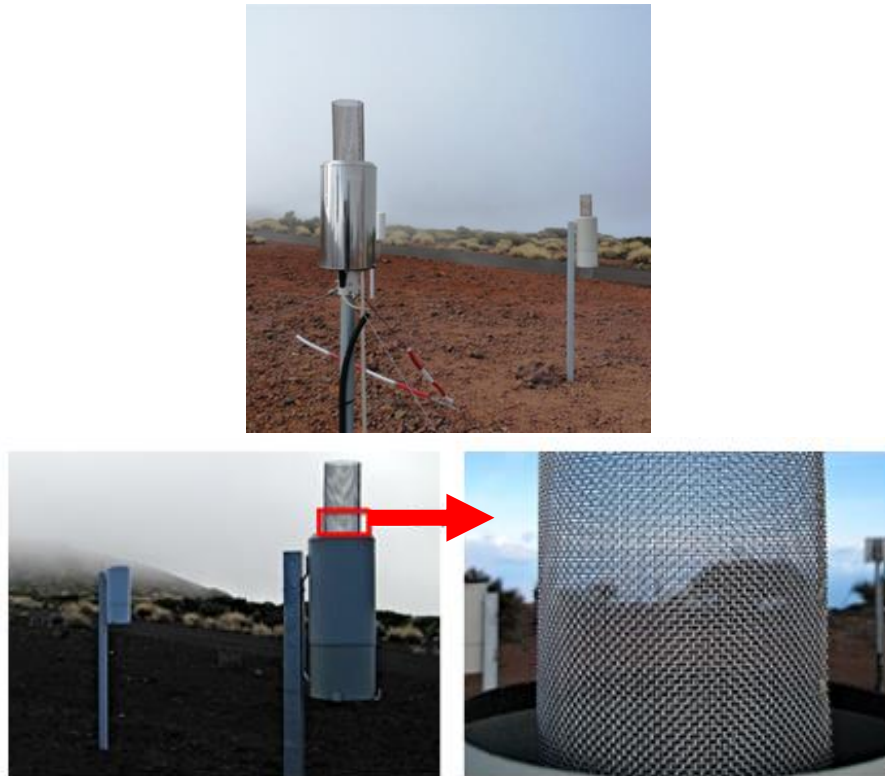


Imagen 16. Los pluviómetros Hellmann establecidos en el entorno del Observatorio Atmosférico de Izaña, en el Parque Nacional de las cañadas de Teide [AEMET 2009].

En cuanto al área de captación de este dispositivo, la sección de recepción del pluviómetro posee un área de captación de reducido tamaño, por este motivo, la norma también establece el uso de un embudo como se ilustra en la imagen 16. Si la zona donde se va a instalar el pluviómetro hay niebla, la norma aconseja colocar una malla cilíndrica para la captación del agua originada por la niebla, tal y como se ilustra en la imagen 16.

Si se cumple la normativa, el pluviómetro puede alcanzar una precisión de la medición de la cuantificación de la precipitación en el orden de $\pm 0,1\text{mm}$.

- **Pluviógrafo:**

Instrumento que registra, de forma continuada y automática, las diferentes medidas de cuantificación de la precipitación obtenidas en el pluviómetro. Este dispositivo va registrando durante un trascurso de tiempo determinado las medidas del pluviómetro y, por lo tanto, su estudio nos indica la evolución y la intensidad de la precipitación. Su registro establece la evolución de la precipitación con respecto al tiempo.

Este dispositivo puede ser analógico o digital.

- **Pluviógrafo analógico:** Existen gran variedad de pluviómetros analógicos, pero los más comunes son el medidor de flotador, el medidor de balanza y el medidor de cubeta basculante, tal como se ilustra en la imagen 18 y se describen en la ilustración 2.

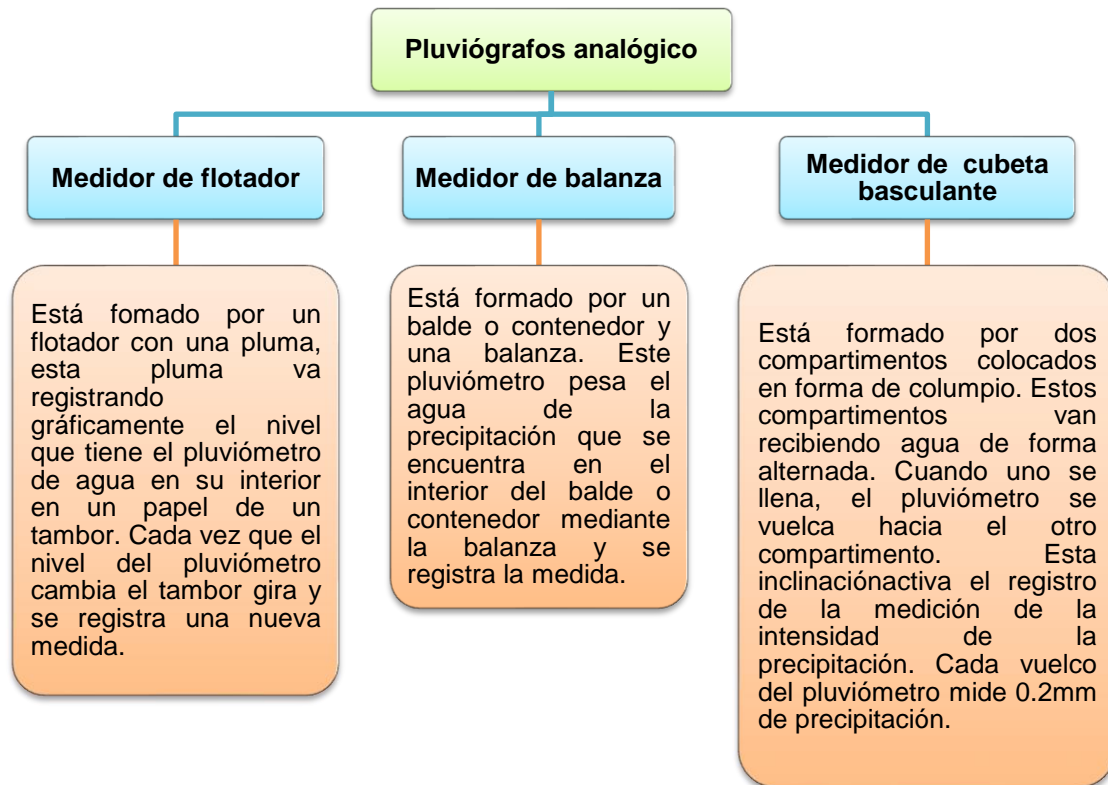


Ilustración 2. Clasificación de los pluviómetros analógicos.



Imagen 17. En la izquierda el pluviómetro de medidor de flotador. En el centro el pluviómetro de medido de balanza. En la derecha el pluviómetro de cubeta basculante.

- **Pluviógrafo digital:** Pluviómetros con sensores que registran de forma digital y automática los datos y los envían mediante conexión inalámbrica. Como se ilustra en la imagen 18.



Imagen 18. Pluviómetro digital.

- **Redes pluviométricas:**

Redes de pluviómetros y estaciones meteorológicas que se instalan para la investigación de la precipitación en zonas de estudio determinadas o de interés. Para un diseño correcto de una red se debe de tener en cuenta:

- El espacio entre estaciones meteorológicas, esto dependerá si el estudio es sobre tormentas genéricas (más espaciado) o el estudio de patrones de las tormentas (menos espaciado).
- Densidad de la red, esto dependerá si las precipitaciones son fuertes (mayor densidad) o débiles (menor densidad).
- Orografía del terreno, relieve, etc.

La OMM establece las recomendaciones para las densidades de las redes pluviométricas. En Canarias, al ser una región montañosa en zona tropical la recomendación es de 100 a 250Km² por estación [OMM 2019].

- **Estaciones climatológicas:**

Son dispositivos cuyo objetivo es la medición y el posterior registro de diversos fenómenos meteorológicos, tales como el viento, la temperatura, la humedad, la insolación, la precipitación y la evaporización.

Todos los dispositivos usados exigen un nivel de normalización, ya que serán usados en diferentes estudios alrededor del mundo.

En este proyecto haremos uso de dos pluviómetros propios, mediciones de radioenlaces y los datos de AEMET que usaremos de referencia para verificar los resultados obtenidos por nuestros algoritmos de aprendizaje.

Los pluviómetros usados en VIMETRI-MAC son los modelos VAISALA WXT530 y VAISALA pluviómetro RG13 que podemos ver en la imagen 19:



Imagen 19. A la izquierda, pluviómetros VAISALA WXT530. A la derecha, VAISALA pluviómetro RG13.

Las características técnicas del pluviómetro VAISALA WXT530 son las siguientes:

- **Tipos de mediciones:** precipitación, presión, temperatura, humedad y viento.
- **Datos de la medición de la precipitación:**

Tabla 2. Datos de la precipitación del pluviómetro modelo VAISALA WXT530

| | |
|--|---|
| Área de captación | 60cm ² . |
| Precipitación | Por acumulación después del último reinicio automático o manual. |
| Resolución | 0.01mm. |
| Precisión para la acumulación diaria. | Mejor al 5%. |
| Duración | Incrementos de 10 segundos desde que se detecta la precipitación. |
| Intensidad | La ejecución tiene un promedio de 1 minuto para pasos de 10 segundos. |
| Rango de la intensidad | 0-200mm. |
| Resolución de la intensidad | 0.1mm/h. |
| Granizo | Cantidad de golpes acumulados contra la superficie de captación. |

Las características técnicas del pluviómetro VAISALA RG13 son las siguientes:

- **Tipos de mediciones:** precipitación.
- **Datos de la medición de la precipitación:**

Tabla 3. Datos de la precipitación del pluviómetro modelo VAISALA WXT530

| | |
|-----------------------------------|----------------------|
| Precisión | Mejor al 2%. |
| Diámetro de apertura | 225mm. |
| Área de apertura | 400cm ² . |
| Capacidad de precipitación | Ilimitada |
| Resolución | 0.2mm. |
| Duración | <100ms |

2.5 Estimación de la precipitación

Las variables fundamentales para la cuantificación de la lluvia son [USAL 2020]:

- Intensidad
- Duración
- Frecuencia

2.5.1 Intensidad

La intensidad representa la cantidad de agua caída por unidad de tiempo. Su expresión es la siguiente [USAL 2020]:

$$I = \frac{P}{t}$$

Donde:

I: Intensidad de la precipitación en mm/h.

P: Altura del agua de precipitación en mm.

t: Duración de la precipitación en horas.

La unidad de litros/m² es equivalente al mm, por ejemplo, un litro en una superficie de 1 m² origina una lámina de agua de 1 mm.

2.5.2 Duración

Duración de la precipitación, es decir, tiempo que transcurre desde el inicio hasta el fin de la precipitación. Esta variable se toma en minutos u horas.

Este parámetro se obtiene del pluviograma o banda pluviográfica.

2.5.3 Frecuencia

La frecuencia representa el número de veces que se repite una precipitación con unas características definidas en cuanto a intensidad y duración en un período de tiempo, generalmente en años. Por su naturaleza, esta frecuencia, así como la duración de las precipitaciones, es difícil de determinar. Se plantea así un problema que afecta a nuestras decisiones técnicas.

En un sistema de seguimiento como el de VIMETRI-MAC necesitamos muestrear de forma continua y, además, con un número de muestras adecuado a cada situación. Por ejemplo, cuando no hay previsión de emergencia se puede muestrear cada 20 a 60 minutos. Sin embargo, cuando hay una situación de (riesgo) de emergencia necesitamos más. Por eso, en nuestro proyecto, uno de los elementos que tenemos en cuenta es muestrear con la suficiente velocidad. Por ejemplo, tomar imágenes cada 5 minutos.

Capítulo 3:

3. La base de datos

En este capítulo se explican detalles técnicos y la organización de la base de datos que en capítulos posteriores nos permitirán explicar y aplicar las bases metodológicas en un aprendizaje supervisado aplicado en este TFM a la detección y la cuantificación de la precipitación de la lluvia. Para ello, se describirá la captación y el almacenamiento de las imágenes suministradas por las cámaras de videovigilancia (propias de VIMETRIC, públicas y privadas) desplegadas por la isla de Gran Canaria. Por ello, estudiaremos dos bloques:

➤ Captación

- Fuentes de los datos recogidos.
- Se detallará los modelos de las cámaras empleadas.
- Ilustración de la ubicación y situación de las cámaras.
- Parámetros técnicos utilizados en la captación de imágenes, tales como, número de imágenes/segundo, calidad de captación, etc.

➤ Organización

- Descripción detallada e ilustración de la organización de las carpetas utilizadas en la base de datos.
- Descripción e ilustración de la nomenclatura de los nombres de los videos.
- Descripción de la cantidad de videos seleccionados para la base de datos con eventos de lluvia y sin lluvia.

3.1 Captación de los datos

Los datos recogidos en la base de datos de este proyecto proceden de dos fuentes principales:

- **Datos recogidos por estaciones meteorológicas:**

Se disponen de unas estaciones meteorológicas que nos aportan los datos recogidos en un archivo tipo csv. Una de estas estaciones es la Agencia Estatal de Meteorología (AEMET) que envía los datos en el marco del Proyecto VIMETRI-Mac que es socia. Esta estación está en la Urbanización Zurbarán, próxima al Campus de Tafira. Este archivo contiene los siguientes parámetros:

- Fecha
- Hora
- Precipitación
- Temperatura en grados centígrados
- Velocidad del viento en Km/h
- Dirección del viento
- Presión en hPa
- Humedad en %
- Otros datos

En este proyecto, sólo necesitaremos los tres primeros parámetros.

- **Datos recogidos por cámaras:**

Lo fenómenos meteorológicos son captados por equipos instalados de red de vigilancia externa:

- **Ubicaciones o bases de los equipos instalados:**

- Base Hospital Rey Juan Carlos I.
- Base ULPGC. Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B.
- Base ULPGC. Edificio de Electrónica y Telecomunicación, pabellón B, planta 2, laboratorio 202.

- **Modelos de cámaras:**
 - **Modelo Grandstream GXV3674-FHD:**

La cámara modelo GXV364_FHD se ilustra en la imagen 20.



Imagen 20. Cámara modelo Grandstream GXV364_FHD.

A continuación, en la imagen 21 se muestran las cámaras ubicadas en la base Hospital Rey Juan Carlos I.



Imagen 21. Cámaras GXV364_FHD de la base Hospital Rey Juan Carlos I.

En la imagen 22 se observa la captura de imagen de visión de cada cámara.



Imagen 22. Visión de las cámaras GXV364_FHD de la base Hospital Rey Juan Carlos I.

En la imagen 23 se muestran las cámaras ubicadas en la base ULPGC, en el Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B.

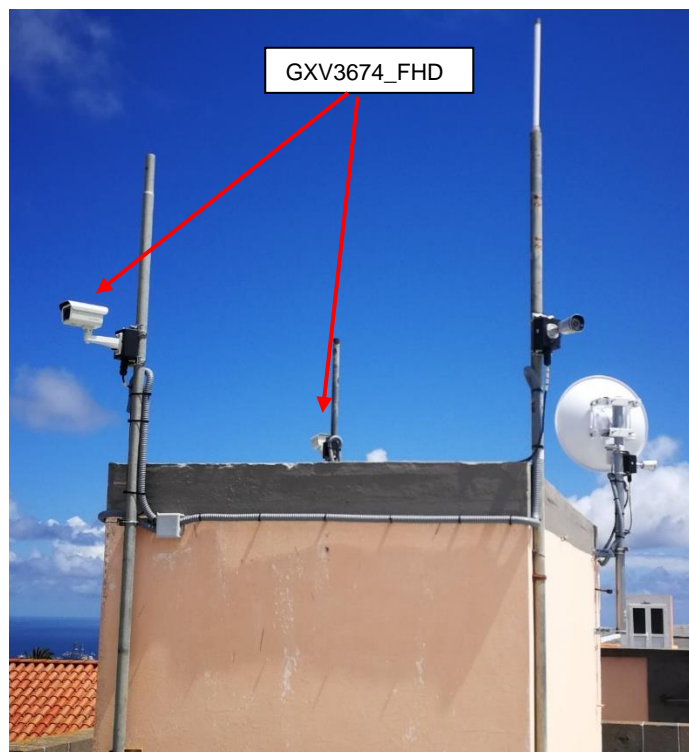


Imagen 23. Cámaras GXV364_FHD de la base ULPGC. Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B.

En la imagen 24 se observa la captura de imagen de visión de cada cámara.



Imagen 24. Visión de las cámaras GXV364_FHD de la base ULPGC. Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B.

La cámara modelo GXV364_FHD tiene las siguientes características técnicas, ilustrada en la tabla 4.

Tabla 4. Características técnicas cámaras GXV364_FHD

| | |
|---|---|
| Compresión de vídeo | H.264, MJPEG |
| Resolución del sensor de imagen | 1/3", 3.1 Megapixel Progressive Scan CMOS, 2048H x 1536V |
| Sensibilidad del sensor de imagen | Modo día/noche. Bajo ruido y baja sensibilidad a la luz. Obturador: 1/10000 - 1/30 segundo |
| Distancia Focal | 2.8 ~ 12 mm |
| Apertura | F1.4, DC-IRIS |
| Angulo de campo | 90° ~ 28° (Campo horizontal) |
| Filtro de corte IR | Si, mecánico |
| Modo día/noche | IR LED cubre hasta 25m |
| Iluminación mínima | 0.05 Lux, 0 lux con IR |
| Receptividad | 1.9V/lux-segundo (550nm) |
| Video máximo admitido | 2048x1536 (15 imágenes) |
| Resolución y velocidad de fotogramas | 1920x1080 (30 imágenes) |
| Velocidad bit | 32 Kbps ~ 8 Mbps velocidad múltiple para vista previa y grabación |
| Entrada de audio | Line-In, 1500 ohms, 0.1Vrms ~ 0.707Vrms, 3.5mm |
| Salida de audio | Line-Out, 600 ohms, 0.707Vrms, 3.5mm |
| Compresión de audio | G.711u/a, AAC |
| Analítica empotrada | Detección de movimiento (hasta 16 áreas objetivo) |
| Pre-/post-alarma de grabación | Si, 3MB |
| WDR (Wide Dynamic Range) | Si (100dB) |
| Instantáneas | Activado por eventos, enviar por correo electrónico / FTP |
| Protocolo de red | TCP/UDP/IP, RTP/RTCP, RTSP, DHCP, DDNS, HTTP, HTTPS, SMTP, FTP, NTP |
| Soporte SIP/VoIP | Si |
| Fotografía de lapso de tiempo | Si |

| | |
|------------------------------------|--|
| Máscara de privacidad | Si, 4 zonas |
| Power over Ethernet (PoE) | IEEE 802.3af, Clase 0 |
| Conexión de cables externos | Red: RJ45, 10M/100M autodetección 3.5mm Line-In 3.5mm Line-Out Power Input |
| Dimensiones | 255mm (L) x 104mm (W) x 187mm (H) |
| Peso | 1.10kg |
| Temperatura/Humedad | En funcionamiento: -20°C ~ +50°C (-4°F ~ 122°F), 10–90% RH (no-condensación) En almacenamiento: -30°C ~ +60°C (-22°F ~ 140°F) |
| Adaptador de alimentación | Salida: 12VDC/1A; Entrada: 100–240VAC, 50–60Hz |
| Caja externa o carcasa | Carcasa de metal resistente a la intemperie IP66 |
| Conformidad | FCC Parte 15, Subparte B Clase B; EN 55022 Clase B, EN 61000-3-2, EN 61000-3-3, EN 55024, EN 60950-1; C-tick AS/NZS CISPR 22, IP66 |

- **Modelo Foscam:**

Las cámaras Foscam utilizadas en este TFM poseen el modelo FI9901EP, y se muestran ubicadas en la base ULPGC, en el Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B (Imagen 25).



Imagen 25. Cámaras FOSCAM de la base ULPGC. Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B.

En la imagen 26 se observa la captura de imagen de visión de cada cámara.



Imagen 26. Visión de las cámaras FOSCAM de la base ULPGC. Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B.

La cámara modelo FOSCAM FI9901EP tiene las siguientes características técnicas, ilustrada en la tabla 5.

Tabla 5. Características técnicas cámaras FOSCAM FI9901EP.

| | | |
|-------------------------|------------------|--|
| Sensor imagen de | Sensor | High Definition Color 1/3" CMOS Sensor |
| | Resolución | 2560 x 1440 Pixels (4.0 Megapixels), 2304 x 1296 Pixels (3.0 Megapixels). 2X Magic Zoom and 6X Digital Zoom |
| | Iluminación | 0 Lux mínimo (con iluminación infrarroja incorporada) |
| | Controles | Control de brillo, contraste y frecuencia de luz. Automáticos y manuales |
| Lentes | Lente | Cristal; Lentes IR-infrarrojas de visión nocturna. f: 2.8 mm, 78° visión diagonal |
| Vídeo | Compresión vídeo | H.264 |
| | Stream | Doble stream |
| | Imágenes/seg. | 30 fps máx. (ajustable a valores inferiores) |
| | Resolución | 2560 x 1440 Pixels (4.0 Megapixels), 2304 x 1296 Pixels (3.0 Megapixels). 2X Magic Zoom and 6X Digital Zoom |
| | Volteo imagen | Vertical / Horizontal |
| | Frecuencia luz | 50Hz, 60Hz o Exterior |
| | Modo infrarrojo | Automático y Manual |
| | Ajustes vídeo | Brillo, Contraste |
| Sonido | Entrada/Salida | Conectores jack estándares 3.5 mm para entrada/salida de audio. Compresión PCM/G.726 |
| Comunicación | Red Ethernet | 10/100 Mbps RJ-45 con PoE IEEE802.3af |
| | Protocolos | HTTP, P2P (acceso vía Internet sin necesidad de abrir puertos), FTP, TCP/IP, UDP, SMTP, DHCP, PPPoE, UPnP, GPRS |
| Almacenamiento | Tarjeta microSD | Ranura para tarjeta microSD hasta 128 Gb, además de microSD normales. Se recomienda clase 10 mínimo. La instalación requiere abrir la cámara. Una vez instalada, no es necesario acceder físicamente a la misma ya que se puede visionar y descargar las grabaciones por software, tanto local como remotamente. |

| | | |
|----------------------------------|--|---|
| Sistemas compatibles | Sistema operativo | Windows XP, Vista, 7, 8, 10; MacOS, iOS, Android |
| | Navegador | IE, Firefox, Chrome, Safari |
| Datos físicos | Luz infrarroja | IR LEDs, Alcance nocturno hasta 20 m |
| | Dimensiones | En mm.: 98 (La) x 65 (An) x 62 (Al) |
| | Orientación | 180° horizontal / 180° vertical |
| | Peso | 500 gr (accesorios incluidos) |
| Alimentación | Alimentación | Adaptador DC 12V/1.0A (incluido). El cable mide 1,5 m. |
| | PoE | Alimentable eléctricamente por PoE (cable ethernet). Estándar IEEE802.3af |
| | Consumo | <5 W |
| Entorno | Temperatura | -20° ~ 55°C (operativa) -20°C ~ 60°(almacenamiento) |
| | Humedad | 10% ~ 80% sin condensación (óptima operativa) 0% ~ 90% sin condensación (almacenamiento) |
| Certificación y Normativa | CE, RoHs, FCC, IP66 (estanqueidad en modelos exterior) | |

- **Equipamiento de cámaras:**

- Herraje de instalación
- Mástil de sujeción
- Cableado
- Armario de exterior con ventilación formado por:
 - Un Switch 8 puertos con POE UBNT-ES-8-150W.
 - Un PSU / Rectificador 48-56V.
 - Un Cuadro eléctrico con magnetotérmico 2P 16A, diferencial 2P 25A 30mA y monitor de tensión con contacto NNNC.
 - Con su correspondiente herraje de instalación, mástil de sujeción y cableado.

- **Equipamiento de recepción de los datos en el laboratorio:**

- Un Router Mikrotik RB2011
- Un Sistema de alimentación ininterrumpida (SAI) de 700VA marca Salicru SPS.900.ONE

3.2 Organización de los datos

La base de datos se llama “Videovigilancia_Meteorologica” y está formada por las siguientes carpetas (imagen 27).

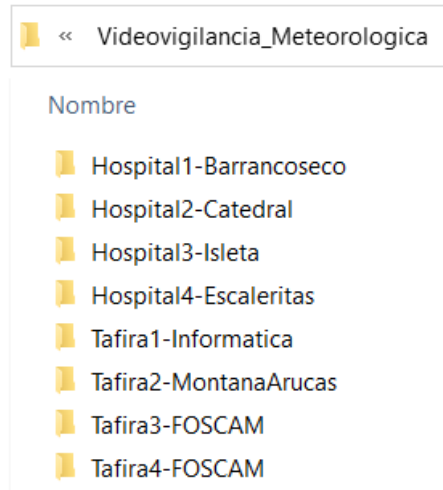


Imagen 27. Contenido de la base de datos.

Como se ilustra en la imagen 27 anterior, está formada a su vez por 8 carpetas. Cada una de estas carpetas pertenece a los datos recogidos por cada una cámara de las que disponemos para el estudio de este proyecto:

- Base Hospital Rey Juan Carlos I:
 - Hospital 1 – Barranco Seco
 - Hospital 2 – Catedral
 - Hospital 3 – Isleta
 - Hospital 4 – Escaleritas

- Base ULPGC. Edificio de Electrónica y Telecomunicación, en la cubierta del pabellón B:
 - Tafira 1 – Informática
 - Tafira 2 – Montaña Arucas
 - Tafira 3 – FOSCAM
 - Tafira 4 – FOSCAM

A su vez, como se observa en la imagen 28, cada una de estas carpetas se dividen en tres subcarpetas. En ellas se recogen los vídeos en función de si contienen lluvia o no, y en caso de haberla su cuantificación entre niveles bajos y altos. La selección se ha hecho por inspección ocular de cada video y con la ayuda de los datos de precipitación (en caso de duda).

- **NL_NN:** No hay datos meteorológicos reales de lluvia y el nivel de precipitación es nulo.
- **SL_NB:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es bajo.
- **SL_NA:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es alto.

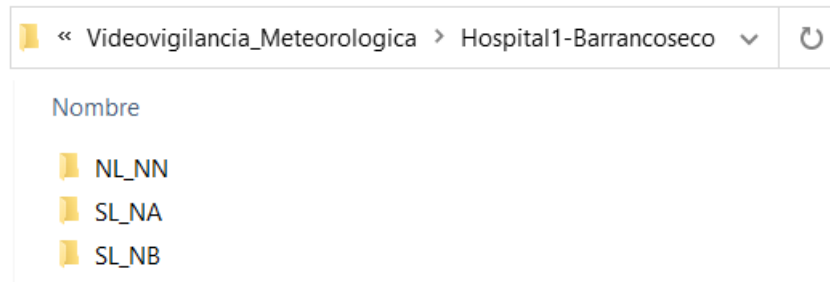


Imagen 28. Subcarpetas para cada carpeta de cada cámara de forma individual.

Y cada una de estas carpetas contiene vídeos de eventos con/sin lluvia según corresponda. En la imagen 29 se muestra el contenido de las carpetas de una de las cámaras, en concreto de la cámara Hospital 1-Barranco Seco:

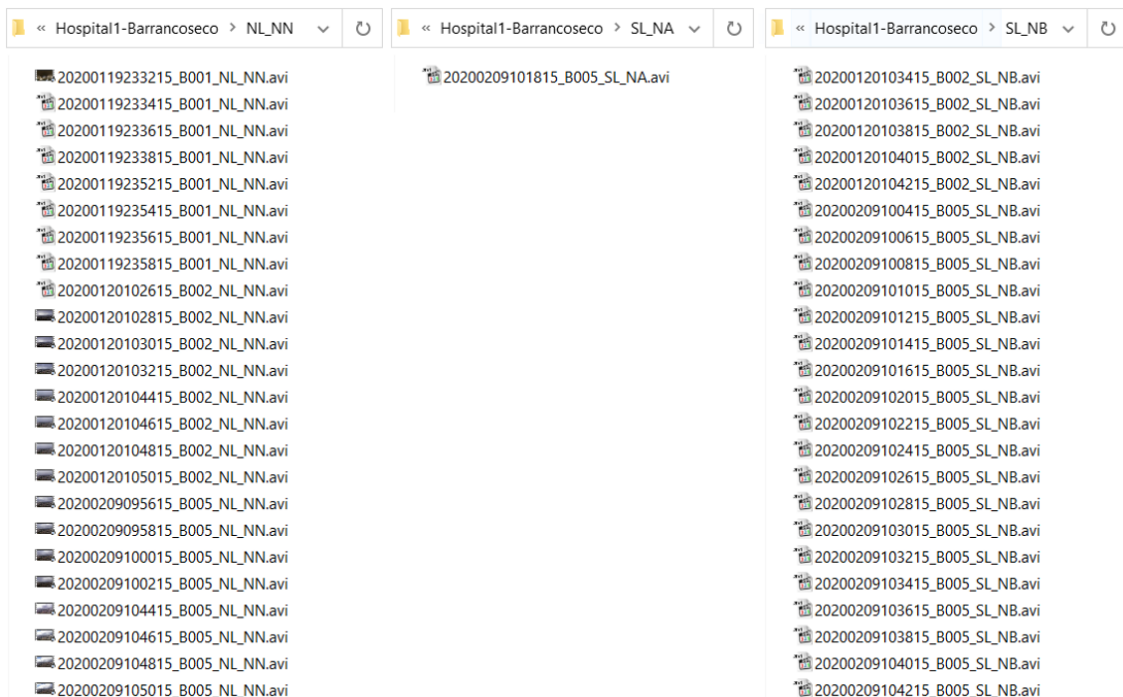


Imagen 29. Contenido de las subcarpetas de la cámara Hospital1-Barranco Seco.

En cuanto a la nomenclatura de los vídeos es la siguiente:

AAAAMMDD hhmss _ BNNN _ CL _ CC

- **AAAAMMDD:** Representa la fecha del vídeo captado, su año, mes y día.
- **hhmss:** Representa la hora del vídeo captado en horas, minutos y segundos.
- **BNNN:** Para saber si un vídeo es consecutivo a otro hemos añadido este parámetro. Cada bloque dispone de un identificador que contiene la letra B seguido de 3 números únicos para cada secuencia de videos.
- **CL:** Este parámetro representa la detección de la precipitación, y viene determinado por dos caracteres:
 - **NL:** Cuando el vídeo no muestra eventos de lluvia.
 - **SL:** Cuando el vídeo si muestra eventos de lluvia.
- **CC:** Este parámetro representa la cuantificación de la precipitación, y viene determinado por dos caracteres:
 - **NN:** No hay eventos de lluvia.
 - **NB:** Si hay un evento de lluvia de nivel bajo, tipo llovizna.
 - **NA:** Si hay un evento de lluvia de nivel alto, tipo chubasco.

Esta base de datos contiene un total de:

- Cantidad de vídeos de eventos con lluvia baja: 418
- Cantidad de vídeos de eventos con lluvia alta: 253
- Cantidad de vídeos de eventos sin lluvia: 251

A continuación, damos más datos del proceso de captación:

- **Formato y compresión de los vídeos:** AVI
- **Período de muestreo entre vídeos:** Se recogen vídeos cada 5 minutos de forma interrumpida. Este período se decidió en los inicios de VIMETRI, durante el diseño del sistema de captación, como solución de compromiso entre un muestreo suficientemente rápido dada la velocidad de cambio de las condiciones meteorológicas y la necesidad de no saturar las bases de datos de almacenamiento.

- **Duración de cada vídeo:** Cada vídeo dura alrededor de 2 segundos y proporciona 30 fotogramas por segundo. Con este tiempo de observación disponemos de suficiente imagen para hacer un aprendizaje máquina adecuado.

Capítulo 4:

4. Procesado y extracción de características de la imagen.

En este capítulo se describirá los ajustes y tareas de normalización de las imágenes a tratar, se implementará algoritmos para la estabilización de las imágenes y, posteriormente, se le aplicará un filtro de gradiente para realzar, previamente, dichas imágenes al aprendizaje máquina.

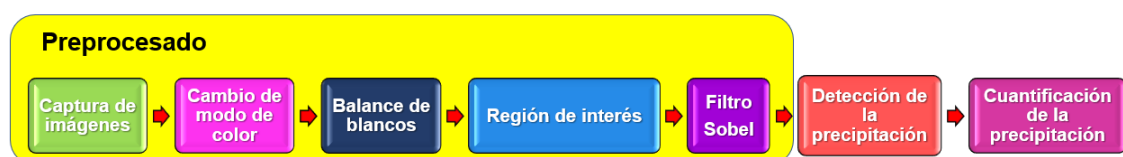


Imagen 30. Diagrama de bloques del pre-procesado de la señal.

4.1 Cambios en el modo de color

Las imágenes captadas de las diferentes cámaras se organizan en una base de datos, donde quedan almacenadas en formato vídeo AVI. A continuación, se extrae cada una de las imágenes que contienen los vídeos almacenados para

ser pre-procesadas (como se observa en la imagen 31). Inicialmente, se le aplicará un filtrado de cambio de modo de color.

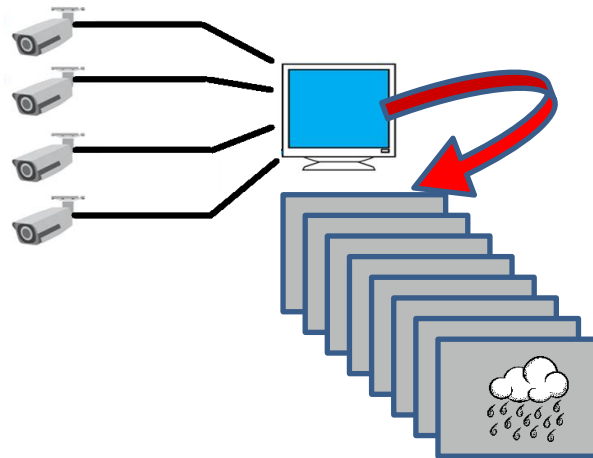


Imagen 31. Captura de las imágenes desde diferentes cámaras.

En este TFM se han realizado dos posibles correcciones de modo de color: Cielab y luminancia.

Entraremos en más detalles de este filtrado en el capítulo 6.

4.2 Filtrado de corrección de blancos

El filtro de corrección de blancos es necesario para normalizar la temperatura de color de las imágenes antes de ser procesadas. Esto se debe a que las imágenes captadas por las diferentes cámaras dependen de la iluminación de la escena, esta iluminación no es la misma durante las 24 horas del día, unas veces puede tener una temperatura de color más fría, es decir, la escena tendría tonos azulados, y otras veces más cálida, tonos más rojizos o amarillentos (en la imagen 33 se observa el cambio de temperatura de la imagen durante el transcurso del día de una de las cámaras). Por ello, como se observa en la imagen 32, se debe normalizar la imagen para poder equilibrar la distribución de las tres componentes de color: Rojo, verde y azul (RGB).

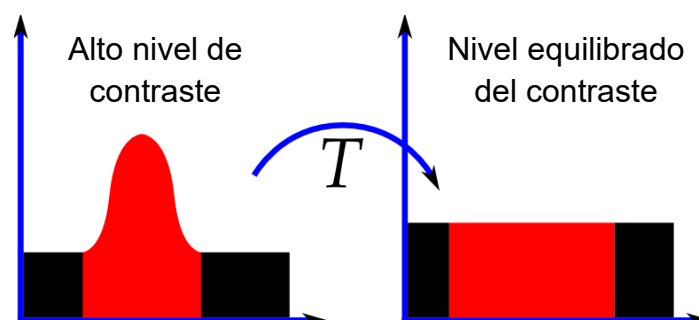


Imagen 32. Normalización de la imagen.

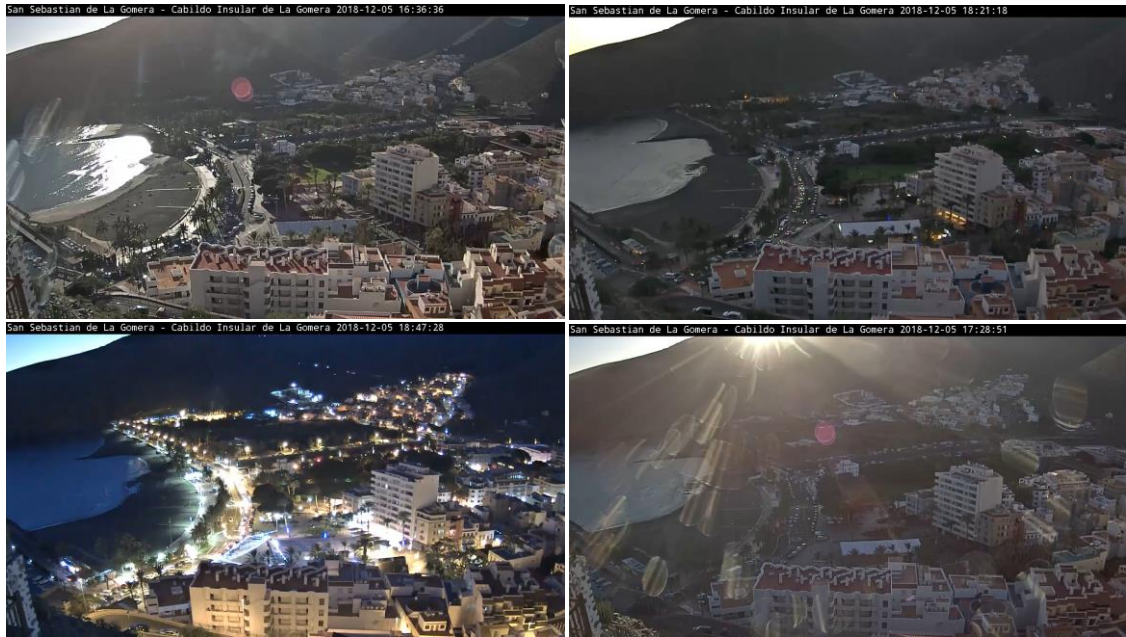


Imagen 33. Imágenes tomadas desde la misma cámara en momentos diferentes del día. Como se observa hay un cambio constante de la temperatura de color de la imagen.

En este proyecto, se han realizado los siguientes pasos para el pre-procesado del filtrado de corrección de blancos (Imagen 34):

1. Cambiamos el modo de color de RGB a LAB, también conocido como CIELAB, este modo de color contiene un espacio de color tridimensional que describe todos los colores que puede distinguir el ojo humano, se descompone en tres componentes:
 - Componente L: luminosidad, variación de negro a blanco.
 - Componente A: croma, variación de rojo a verde.
 - Componente B: croma, variación de azul a amarillo.
2. Se realiza un filtrado de corrección independiente para cada una de estas componentes.
3. Se vuelva a cambiar el modo de color de LAB a RGB.

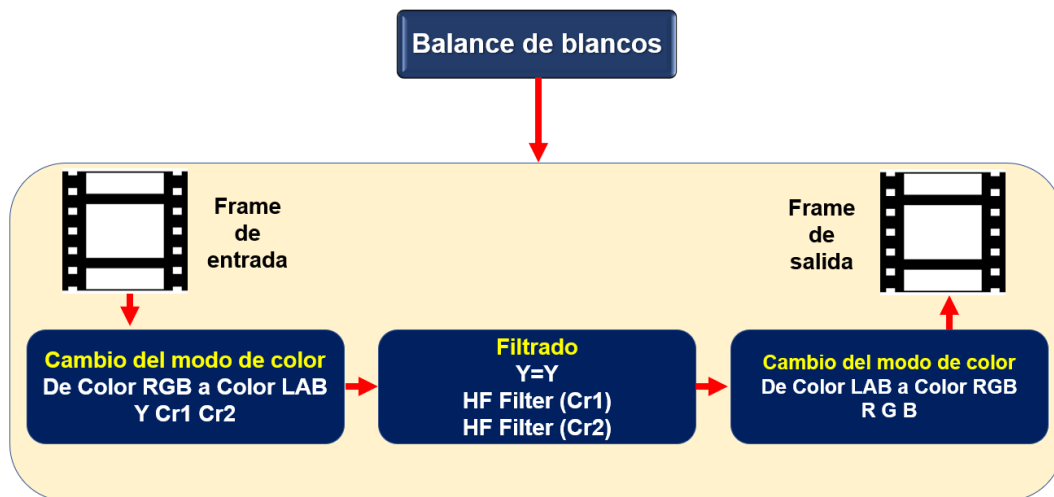


Imagen 34. Diagrama de bloques del balance de blancos.

4.3 Región de interés

Para un mejor estudio de la imagen se le ha aplicado una sectorización para seleccionar una región de interés (Region of Interest, ROI). En cada imagen se pueden seleccionar 9 posibles regiones de interés como se muestra en la siguiente imagen 35.

Después de un análisis exhaustivo de las imágenes de todas las cámaras hemos comprobado que podemos trabajar con 9 regiones de interés posibles, que nos permiten dos cosas:

- Focalizar zonas de mayor contraste de presencia de ruido.
- Identificar las zonas geográficas de mayor interés.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |



Imagen 35. Posibles regiones de interés en las que hemos dividido la imagen. Estas imágenes pertenecen a la cámara Tafira3-FOSCAM y ROI número 8.

Con este filtrado podemos seleccionar de forma estratégica la sección de imagen donde realizaremos el estudio de la detección y cuantificación de la precipitación de la lluvia.

Este filtrado nos permite seleccionar las zonas que presentan componentes de alta frecuencia mejorando el resultado posterior del filtrado de Sobel [Godoy 2019].

4.4 Filtrado de gradiente. Algoritmo Sobel

Mediante este algoritmo se aplica un operador que permite estimar el gradiente de la intensidad de una imagen en cada píxel, esto facilita la detección de cambios dentro de una imagen con respecto a otra. Por lo tanto, se trata de un descriptor asociado a las altas y bajas frecuencias, es decir, un filtro que detecta los bordes de una imagen. Con este descriptor buscamos cambio asociados a la lluvia. Por ejemplo, la presencia de gotas o trazas de ellas al caer y quedan en la imagen.

De forma general, los pasos que realiza este filtrado son los siguientes [Julio Pardo 2020] [Sobel 1973] [Duda 1973] [Bronte 2009]:

1. Se observan los píxeles de la imagen. Para cada píxel, queremos ver los píxeles que lo rodean (Imagen 36).

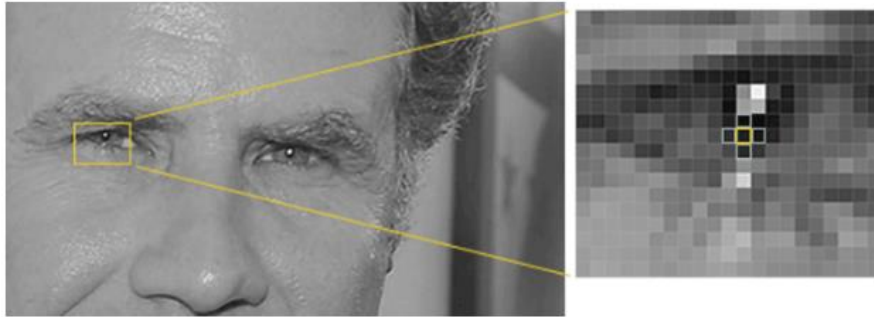


Imagen 36. Zoom de los píxeles que forma la imagen.

2. Observamos el pixel actual con los píxeles en su vecindad. Y determinamos el grado de oscuridad entre ellos, y luego dibuja una flecha que muestre en qué dirección se vuelve más oscura la imagen. Cada píxel se reemplaza por una flecha. Estas flechas se llaman gradientes y muestran el flujo de claro a oscuro en toda la imagen (Imagen 37).



Imagen 37. Grado de oscuridad de un pixel.

3. La imagen original se convierte en una representación de histogramas de gradientes orientados que captura las principales características de la imagen independientemente de los brillos de la imagen (Imagen 38).

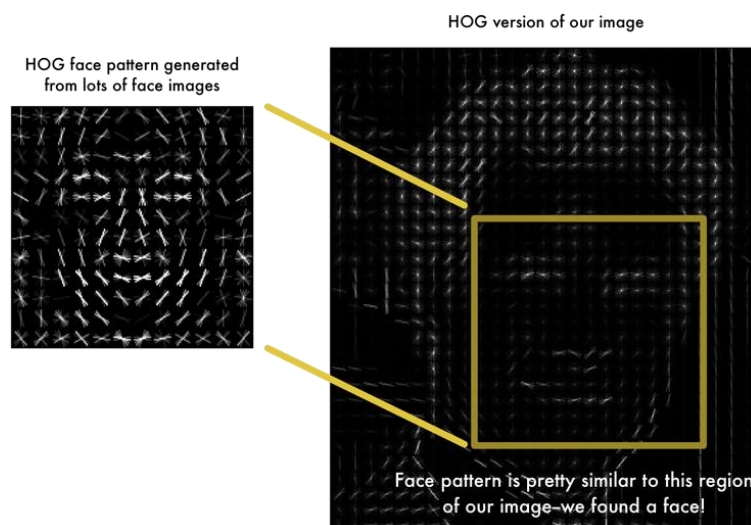


Imagen 38. Ejemplo de los gradientes de la imagen.

Veamos el filtrado de Sobel desde un punto de vista más básico, que expresa cómo lo hemos implementado en este proyecto [Wikipedia 2021]. El operador Sobel calcula el gradiente de la intensidad de una imagen en cada punto (píxel). Es decir, con el gradiente se indica la dirección de la máxima variación de una función. Así, para cada punto, este operador da la magnitud del mayor cambio posible, la dirección de este y el sentido desde oscuro a claro.

El gradiente de una función de dos variables (en este caso, la función de intensidad de la imagen) para cada punto es un vector bidimensional cuyos componentes están dados por las primeras derivadas de las direcciones verticales y horizontales. Recordemos que un borde es una discontinuidad, y para detectar el mismo se debe obtener la derivada de la función de la imagen.

En la imagen 39, se observa el gradiente representado como el vector \overline{uv} , con sus dos componentes:

- **Componente vertical:** se obtiene con el cálculo de la derivada en “y” del punto, en el TFM la denominamos M_y multiplicada por un factor $\frac{1}{2+k}$.
- **Componente horizontal:** se obtiene con el cálculo de la derivada en “x” del punto, en el TFM la denominamos M_x multiplicada por un factor $\frac{1}{2+k}$.

Donde el valor de k depende del operador que utilicemos, en este caso, al ser el operador Sobel, se debe de usar el valor de $K=2$. Este factor garantiza que el valor resultante este siempre entre $[-255, 255]$. Ya que estos filtros pueden producir valores negativos que son causa de problemas por su difícil interpretación y representación como imágenes de niveles de gris.

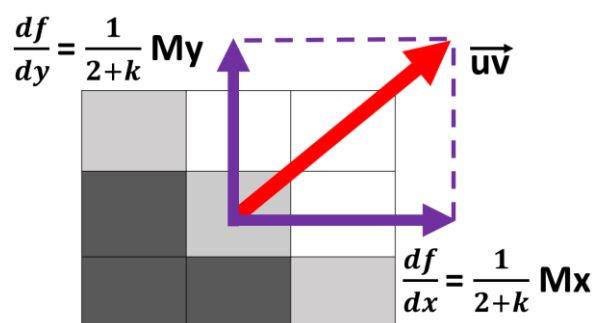


Imagen 39. Gradiente en un punto o píxel.

Para cada punto o píxel de la imagen, se puede calcular el ángulo y la magnitud del gradiente, es decir, del vector \overline{uv} :

- **Ángulo:** Indica dirección del gradiente o máxima variación de la función de un punto o píxel (Ecuación 1).

$$\theta = \tan^{-1}\left(\frac{\frac{df}{dy}}{\frac{df}{dx}}\right)$$

Ecuación 1. Cálculo del ángulo.

- **Magnitud:** Indica la magnitud del gradiente o intensidad de la variación de la función de un punto o píxel (Ecuación 2).

$$\text{Magnitud} = \nabla f = \sqrt{\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2}$$

Ecuación 2. Cálculo de la magnitud.

Las máscaras de Sobel obtenidas son las expresadas en la imagen 40.

$$M_x = \begin{bmatrix} 1 & 0 & -1 \\ k & 0 & -k \\ 1 & 0 & -1 \end{bmatrix} \quad M_y = \begin{bmatrix} 1 & k & 1 \\ 0 & 0 & 0 \\ -1 & -k & -1 \end{bmatrix}$$

Imagen 40. Máscaras de convolución del operador Sobel para el gradiente fila (Mx) y el gradiente columna (My).

Como en el operador Sobel K=2, las máscaras quedarían como en la imagen 41.

$$M_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Imagen 41. Gradiente fila y columna aplicando el valor de K=2 del operador Sobel.

Una vez calculadas las derivadas de las direcciones “X” e “y”, implementamos una doble convolución en la que primero aplicamos un filtrado horizontal y, sobre el resultado aplicamos uno vertical. Como resultado obtenemos Cx e Cy. El objetivo es extraer características de gradiente en horizontal y vertical, simultáneamente (Imagen 42).

```
Cx = conv2(I2,Mx); % Filtrado en X
Cy = conv2(I2,My); % Filtrado en Y
```

Imagen 42. Código de matlab donde realizamos la doble convolución.

Para finalizar el proceso, calculamos la magnitud del gradiente mediante el cálculo del valor máximo absoluto de las derivadas anteriores (Ecuación 3) y no como la expresada en la Ecuación 2 por simplicidad en el proceso de cálculo.

$$I(x,y)=\max\{|Mx|,|My|\}$$

Ecuación 3. Cálculo alternativo para calcular la magnitud del gradiente.

Este filtrado tiene como ventaja que produce un alisamiento en la imagen, dado el comportamiento ruidoso que presentan las estimaciones basadas en las derivadas. Para explicar esta afirmación observaremos este comportamiento en las máscaras. Se le aplica a la imagen dos máscaras sucesivas, una sobre las filas y otra sobre las columnas. La primera se le denomina máscara diferencial cuyo valor es el (1,0,-1) y se aplicará sobre cada fila/columna y, continuación, al resultado se le aplica la máscara de alisamiento, cuyo valor es (1,2,1) sobre cada fila/columna (Imagen 43).

$$\begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} = \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline -1 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array}$$

Imagen 43. Máscara diferencial y máscara de alisamiento.



Imagen 44. En esta imagen se ilustra las imágenes antes y después del filtro Sobel. Pertenecen a la cámara Tafira3-FOSCAM y ROI número 8. Introducimos en la parte superior una imagen en blanco y negro y debajo la misma imagen a color.

El resultado de la imagen 44 muestra cómo de abrupta o suavemente cambia una imagen en cada punto analizado y, en consecuencia, cuán probable es que este represente un borde en la imagen y, también, la orientación a la que tiende ese borde. En la práctica, el cálculo de la magnitud (probabilidad de un borde) es más fiable y sencillo de interpretar que el cálculo de la dirección y sentido, y por eso lo hemos adoptado en este trabajo. No olvidemos que nuestro interés es detectar la presencia de gotas de lluvia cayendo que se superponen a una imagen donde no las hay si no llueve.

Capítulo 5:

5. Marco metodológico para la clasificación de las imágenes.

En la actualidad, el aprendizaje máquina juega un papel muy importante en múltiples áreas de la ciencia, industria e incluso de las finanzas. Este tipo de aprendizaje ayuda a resolver con mayor eficacia y rapidez problemas complejos, tales como [Hastie 2009]. Veamos ejemplos de campos científicos donde se aplica.

- En el área médica se utiliza para:
 - Identificador de factores de riesgos: Identificar si un paciente es probable que esté desarrollando un cáncer.
 - Predictivo: Predecir si un paciente puede tener un ataque al corazón.
 - Estimador: Estimar las cantidades de azúcar en sangre de un paciente diabético.
- En el área financiera:
 - Predictivo: Predecir el precio de una acción financiera basándose en el estudio de la empresa y su plan estratégico.
- En el área industrial
 - Estimador: Estimar las existencias de una empresa en los próximos meses para organizar un mejor control del inventario y anticiparse a futuros problemas.

En este TFM nos introducimos en el problema complejo de la monitorización meteorológica, y hemos empleado el aprendizaje máquina para detectar y cuantificar la precipitación de la lluvia basándonos en imágenes recibidas desde cámaras de vigilancia desplegadas por la isla de Gran Canaria.

5.1 Generalidades del aprendizaje máquina

La Inteligencia Artificial es una disciplina que, dicho en sentido amplio, busca la creación de máquinas que puedan imitar comportamientos inteligentes.

Dentro de la inteligencia Artificial se enmarca el concepto de Machine Learning (ML) o Aprendizaje Máquina, que busca cómo dotar a las máquinas de capacidad de aprendizaje. En cuanto a la forma de aprender podemos distinguir tres categorías principales según el paradigma de aprendizaje:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.
- Aprendizaje reforzado.

En el siguiente apartado explicaremos con detalle cada una de estas categorías. Adelantamos que en este TFM usaremos el aprendizaje supervisado pues disponemos de una base de datos con videos que recogen, debidamente etiquetados, eventos de lluvia y no lluvia.

Dentro de Machine Learning existen diferentes técnicas para resolver diferentes tipos de aplicaciones, por ejemplo [Goodfellow 2016]: Los árboles de decisión, modelos de regresión, modelos de clasificación, técnicas de clusterización, redes neuronales, entre otras técnicas.

Una de las técnicas más importantes a día de hoy son las redes neuronales, estas se basan en redes que pueden aprender de forma jerarquizada, toda la información se va adquiriendo por niveles. Los primeros niveles se aprenden conceptos básicos y los últimos niveles conceptos más complejos. Cuantos más niveles más complejidad puede aprender la red neuronal. Los algoritmos que desarrollan estas capas o niveles se le denomina Deep Learning o Aprendizaje Profundo [Hastie 2009].

5.2 Formas de aprendizaje

Como se ha comentado anteriormente Machine Learning o Aprendizaje Máquina se clasifica en tres categorías según el paradigma de aprendizaje, es decir, según los mecanismos que permiten procesar toda aquella información nueva que percibe la red neuronal para transformarla en conocimiento: [Hastie 2009]

- **Aprendizaje supervisado:** Se basa en analizar la relación entre unas variables de entrada y unas variables de salida. A los algoritmos diseñados se les enseña cuál debe ser el resultado esperado para un determinado valor de entrada. Para ello, se supervisa su entrenamiento mostrándole múltiples ejemplos de entrada y los resultados de salida esperados, de esta forma la red neuronal analiza cual es la relación entre la entrada y la salida y aporta los resultados de forma exitosa.

Este aprendizaje utiliza una clasificación previa en la entrada, a cada clasificación se le denomina clase, categoría o etiqueta descriptiva. Cada clase tiene asociada una salida concreta, por ejemplo, en la imagen 45 se observa que disponemos de tres clases en la entrada con las que vamos a entrenar al sistema, estas son: la clase manzana, la clase plátano y la clase mango, y a su vez, tiene una salida concreta, que es la identificación de la fruta introducida.

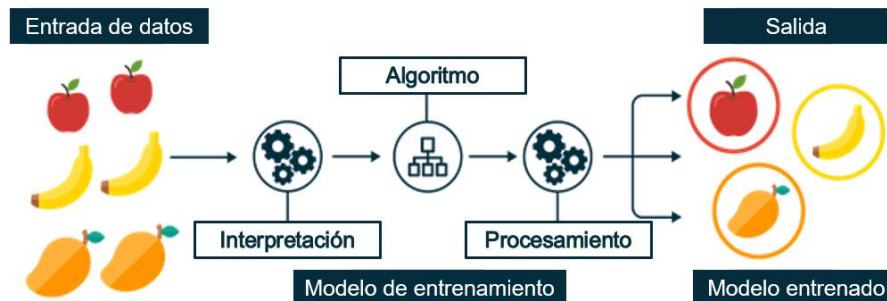


Imagen 45. Diagrama de bloques del aprendizaje supervisado.

En el modelo de entrenamiento, según el objetivo perseguido hace una regresión o una clasificación de la salida. Por lo tanto, se pueden distinguir dos tipos de tareas:

- **Tarea de regresión:** Aprendizaje que predice resultados cuantitativos. La tarea de predicción se puede realizar mediante diferentes modelos, uno de los más usados es el siguiente:
 - **Modelo de regresión lineal:** Inicialmente disponemos de un vector de entradas $X^T=(X_1,X_2,\dots,X_p)$ y un vector de salida $Y=\beta_0+\sum_{j=1}^p(X_j \cdot \beta_j)$, donde β_0 es el sesgo de la máquina de aprendizaje.
- **Tareas de clasificación:** Aprendizaje que predice resultados cualitativos. Es el caso aplicado en este TFM, si bien mediante cuantificación por niveles de lluvia hacemos una forma de regresión.

Como se observa en la imagen 46, un ejemplo de una tarea de predicción es predecir el valor de una casa en los próximos meses según sus características actuales. Y, un ejemplo de clasificación, es predecir si lloverá o no en los próximos días.

Aprendizaje supervisado

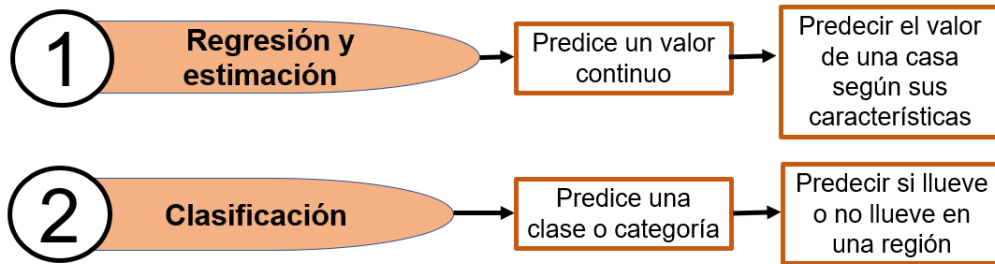


Imagen 46. Modelos del aprendizaje supervisado.

- **Aprendizaje no supervisado:** Este aprendizaje adquiere los conocimientos únicamente de los datos aportados en la entrada sin indicarle al sistema cual es el resultado que esperamos obtener. Para ello, el sistema genera un agrupamiento o clusterización de los datos de la entrada buscando los patrones de similitud de los datos de entrada y genera una respuesta para cada entrada. Esta forma de aprendizaje tiene una dificultad intrínseca, esto es, no sabe si la respuesta generada es la correcta.

Como se observa en la imagen 47 existen diferentes modelos del aprendizaje no supervisado.

Aprendizaje no supervisado

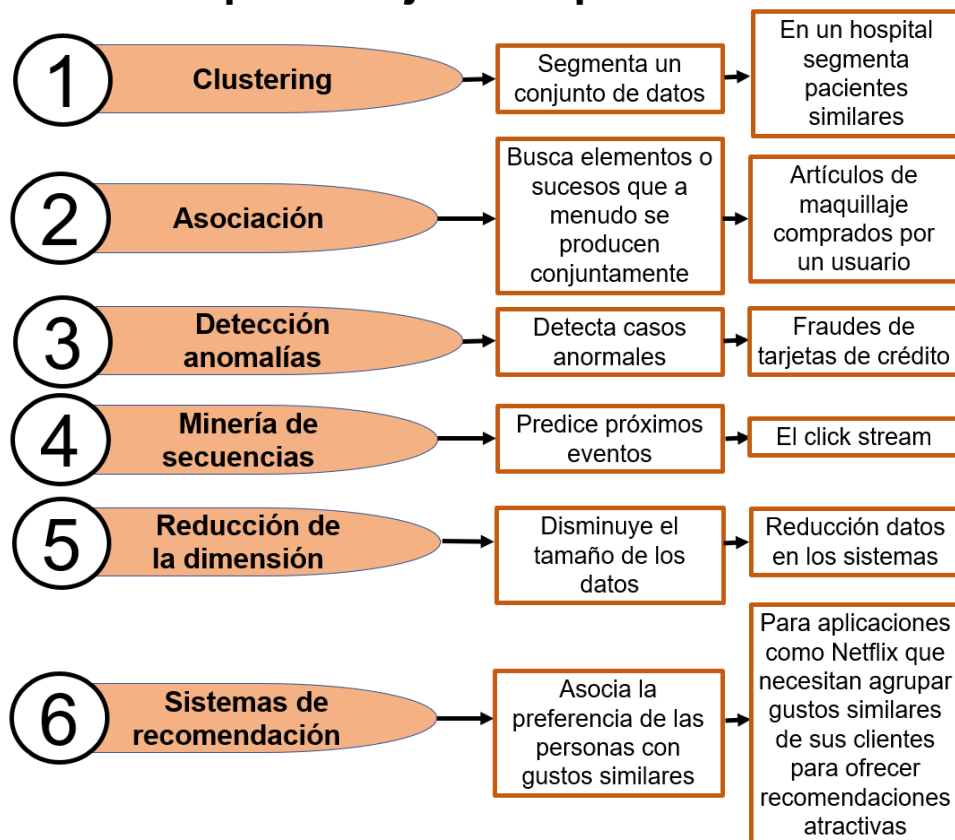


Imagen 47. Modelos del aprendizaje no supervisado.

- **Aprendizaje reforzado:** Este tipo de aprendizaje se rige por la psicología conductista, es decir, por leyes que modifican nuestra conducta voluntaria. Este aprendizaje tiene como objetivo conseguir que una máquina realice una acción concreta y la repita. Para ello, realiza un proceso donde se le otorga al usuario un premio o recompensa tras realizar dicha acción. La máquina al recibir la recompensa aumenta su probabilidad de repetir esa acción premiada y descarta aquellas que no recibe recompensa. Este aprendizaje se observa en la imagen 48.

Aprendizaje por refuerzo

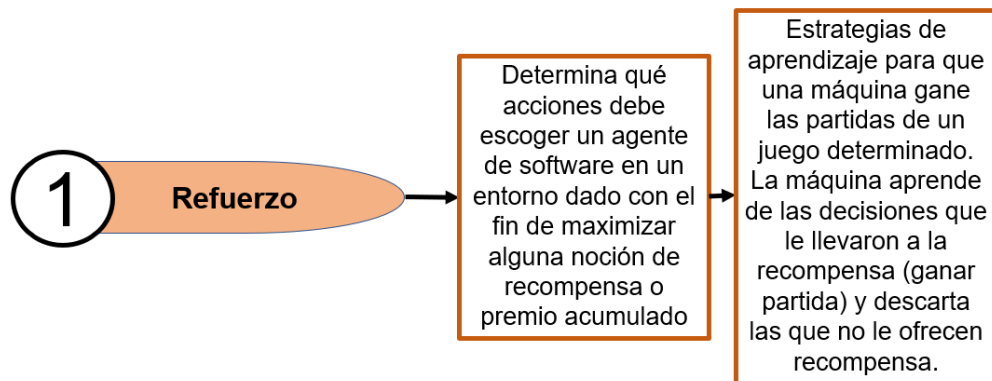


Imagen 48. Modelos del aprendizaje por refuerzo.

5.3 Conceptos básicos de redes neuronales

Las redes neuronales son modelos matemáticos inspirados en el comportamiento de las neuronas. Las neuronas son células que forman la unidad funcional y estructural del sistema nervioso del cuerpo humano. Su función es procesar y transmitir señales que recibe del medio para, posteriormente, generar respuestas acordes a ellas mediante impulsos eléctricos por el cuerpo humano. El sistema que forma el cerebro humano es un sistema complejo, ya que posee 10^{11} neuronas y, por lo tanto, más de 500 billones de conexiones neuronales. Realizar un sistema tan complejo de forma artificial no es una tarea fácil y requiere de ordenadores y tecnologías potentes [Kriesel 2005].

Para entender una red neuronal es importante estudiar previamente las partes en las que se descompone las neuronas biológicas. Se divide en varias partes fundamentales (imagen 49):

- **Soma o cuerpo neuronal:** está formado por el cuerpo central y contiene el núcleo celular y el citoplasma de la neurona. Es el lugar donde tiene lugar todos los procesos metabólicos neuronales. Su función es garantizar la correcta transmisión de las señales eléctricas.

- **Dendritas:** Ramificaciones terminales. Su función se basa en recibir los neurotransmisores generados por las neuronas más próximas.
- **Axón:** Ramificación en el lado contrario a las dendritas. Su función se basa en conducir el impulso eléctrico hasta las conexiones de sinapsis, donde la información es transmitida a la siguiente neurona.
- **Sinapsis:** Conexión entre neuronas. Recibe, procesa y transmite señales a otras neuronas mediante unas moléculas denominadas neurotransmisores.
- **Núcleo celular:** Se encuentra en el interior del soma y en su interior se encuentra el ADN de la neurona.
- **Vaina de mielina:** Sustancia formada por proteínas y grasas que rodea al axón. Su función es facilitar el viaje del impulso eléctrico a una velocidad adecuada.
- **Nódulos de Ranvier:** Separación de micrómetro de longitud entre los packs de la vaina de mielina. Su función es recibir electrolitos de sodio y potasio para hacer que la señal eléctrica viaje correctamente.
- **Cono axónico:** Región del cuerpo de la neurona que se estrecha para dar lugar al axón.

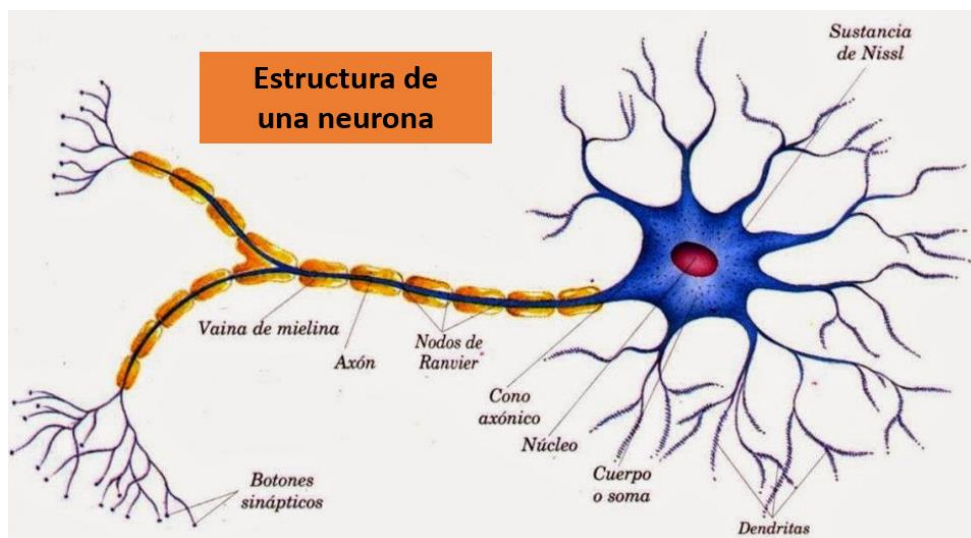


Imagen 49. Estructura de una neurona.

El primer modelo matemático de una neurona artificial se presentó en el año 1943 por Warren McCulloch (psiquiatra y neuroanatomista) y Walter Pitts (matemático). Esta neurona realizaba tareas simples, pero, en la actualidad, las

redes neuronales han ido evolucionando y aumentando su nivel de complejidad. El modelo de neurona presentado se ilustra en la imagen 50, donde podemos observar las nomenclaturas de la neurona biológica frente a la neurona artificial.

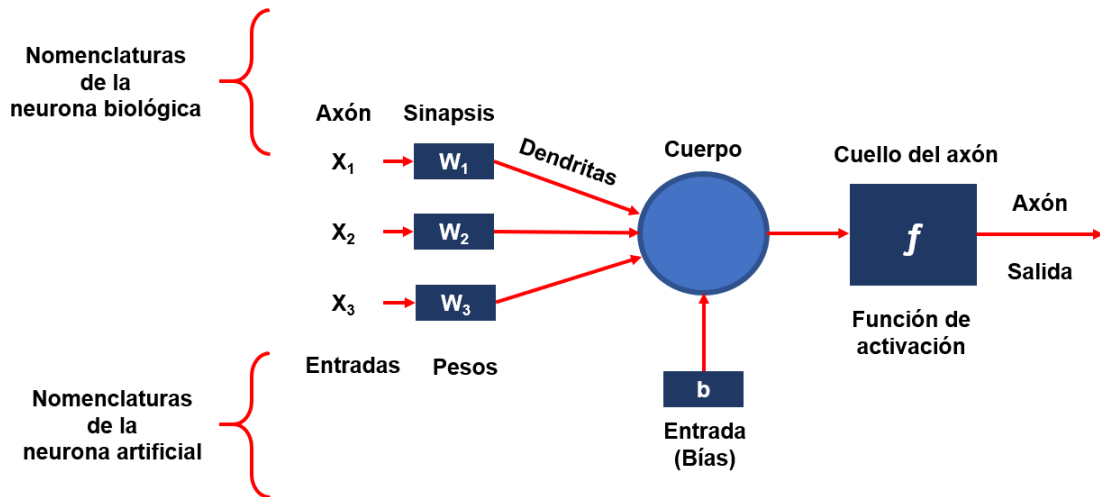


Imagen 50. Primer modelo matemático de una neurona artificial.

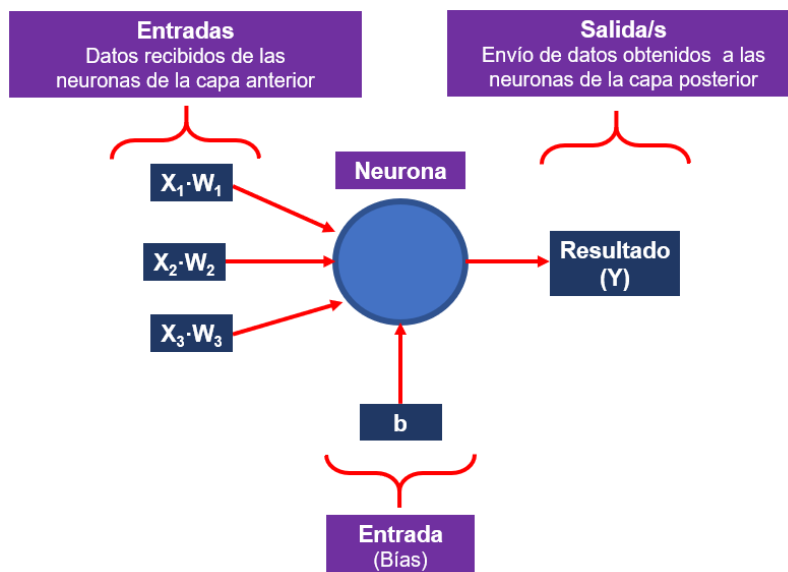


Imagen 51. Entradas y salidas de una neurona que forma la red neuronal.

Las neuronas reciben tres tipos de datos de información de entrada y un dato de salida (imagen 51).

Los datos de entrada:

- **Datos de entrada:** Conjunto de datos de entrada. Se denominan " X_1, \dots, X_n ".

- **Peso de la conexión o peso sináptico:** El peso es un valor que nos indica cuál de las interconexiones con las neuronas de la capa anterior tiene mayor prioridad o importancia. Se denominan “ W_1, \dots, W_n ”.
- **Bias o sesgo:** Este parámetro indica qué neuronas se deben de activar con mayor facilidad frente a otras. Se denominan “ b ”. Para ello, dispone de una función de activación dentro de la neurona. Esta función está basada en la siguiente expresión $f = \sum_{i=1}^n (X_i \cdot W_i) + b$. Si el resultado de esta expresión supera un determinado umbral permite que la neurona se active y envíe sus datos de salida a las siguientes neuronas que forman la capa posterior, en el caso contrario desactivaría la neurona y no le permitiría enviar los datos a las neuronas de la próxima capa.

El dato de salida:

- **Datos de salida:** Resultados obtenidos en la neurona. Si el parámetro bias permite la activación de la neurona este dato se enviará a las neuronas de la capa posterior. Se denomina “ Y ” y viene expresado por la siguiente ecuación $Y = \sum_{i=1}^n (X_i \cdot W_i)$.

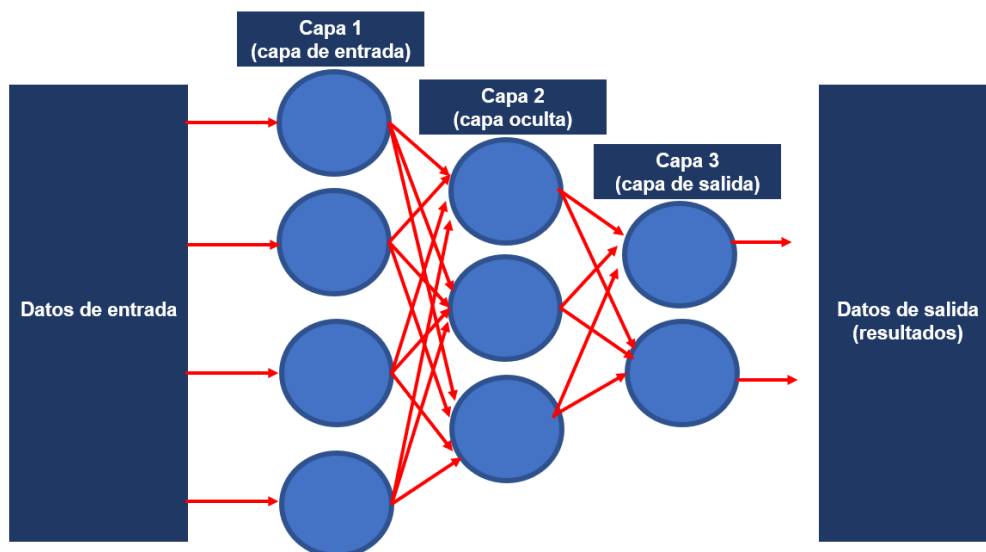


Imagen 52. Red neuronal.

En cuanto a la red neuronal está formada por tres partes (imagen 52):

- **Los datos de entrada:** inicialmente le entregaremos unos datos de entrada de entrenamiento. Al finalizar el entrenamiento le entregaremos unos datos para la predicción de la red neuronal.
- **Las capas:** Cada capa realiza una tarea determinada y está formada por neuronas. Cada neurona está conectada con todas las neuronas de la capa posterior, por lo tanto, están todas interconectadas entre ellas. Se disponen de tres tipos de capas:

- **La capa de entrada:** Capa formada por neuronas que se encargan de recibir todos los datos de entrada y son las encargadas de recopilar la información recibida y enviarla a la siguiente capa.
 - **Las capas ocultas o intermedias:** Formada por una o varias capas y, estas a su vez, por neuronas que se encargan de realizar todo el aprendizaje del algoritmo de la red neuronal.
 - **La capa de salida:** Capa formada por neuronas encargadas de recibir los datos de las capas ocultas para posteriormente tomar una decisión y ofrecer un resultado de salida.
- **Los datos de salida:** Recepción de los resultados obtenidos de la red neuronal mediante la capa de salida.

5.4 Redes neuronales convolucionales

Las redes neuronales tradicionales presentan problemas en el aprendizaje automático cuando la dimensión de los datos es alta (por ejemplo, datos tales como imágenes, vídeos, etc), ya que requiere más carga computacional y se vuelve un proceso inviable. Es por ello, que se ha recurrido al diseño de unas nuevas redes neuronales para solventar este tipo de problema denominadas redes neuronales convolucionales (Convolutional Neural Network, CNN) [Goodfellow 2016].

Como se observa en la imagen 53, las redes neuronales convolucionales tienen una estructura similar a las redes neuronales tradicionales, pero incluye dos algoritmos nuevos en las capas intermedias. Estos algoritmos se denominan algoritmo convolucional y algoritmo Max-Pooling o Average-Pooling. El objetivo de estos algoritmos es reducir la dimensión de los datos según van avanzando a lo largo de la red o estructura. Este sistema se basa en entrenar cada parte de la red para que realice una única tarea, de esta forma se puede reducir el número de capas ocultas o intermedias y el entrenamiento reduce su tiempo de desarrollo significativamente.

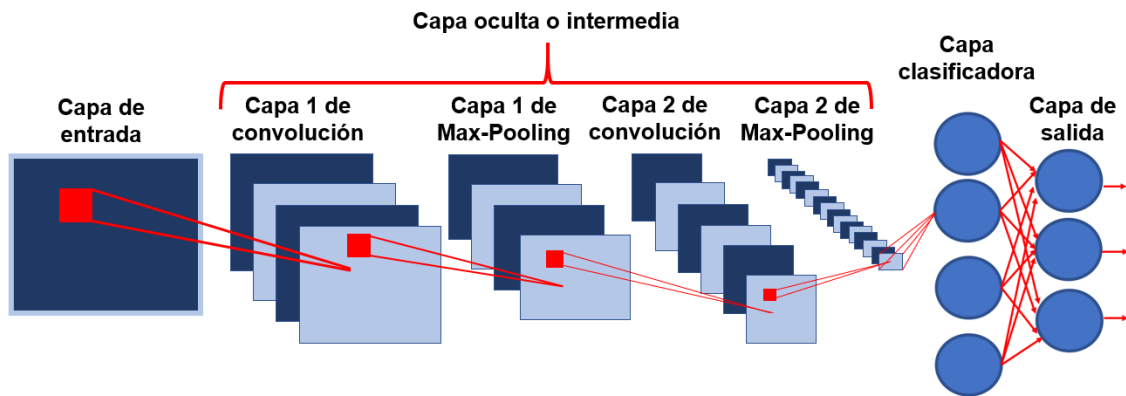


Imagen 53. Estructura de una red neuronal convolucional.

La red neuronal convolucional consta de 4 capas fundamentales:

- **Capa de entrada:** Capa formada por neuronas que se encargan de recibir todos los datos de entrada (si el dato de entrada es una imagen, recibe todos los píxeles de dicha imagen) y son las encargadas de recopilar la información recibida y enviarla a la capa intermedia u oculta.
- **Capas ocultas o intermedias:** Capas formadas por neuronas que se encargan de realizar todo el aprendizaje del algoritmo de la red neuronal. Según el algoritmo de aprendizaje, se divide en dos tipos de capas:
 - **Capa convolucional:** Esta capa realiza las operaciones de productos y sumas para generar un mapa de características. Cada una de estas características corresponde con una ubicación del filtro de la imagen original.

Como se observa en la imagen 54, la imagen de entrada a la capa convolucional se va a ir recorriendo con un filtro, de paso predeterminado, y se va a ir generando una imagen de salida. En cuanto a la imagen de salida siguiente.

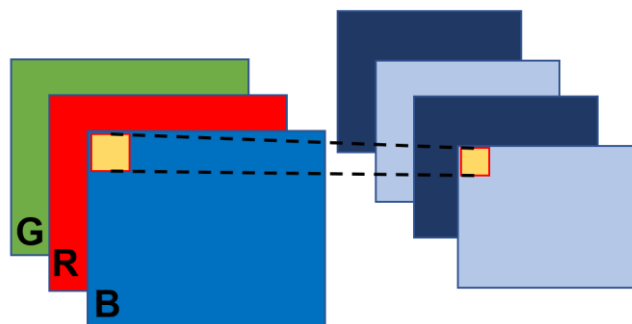


Imagen 54. Capa convolucional.

- **Las dimensiones del alto y ancho:** dependerá del tamaño del filtro que hemos predeterminado en la capa convolucional. El objetivo es reducir estos dos parámetros.
- **La dimensión de la profundidad:** depende del número de filtros que le vamos a realizar a la imagen en cada capa de las convoluciones. Por ejemplo, en la imagen 55 se han realizado tres filtrados diferentes a la imagen, podría ser un filtro para detectar bordes, otro para detectar sombras y otro para detectar contrastes, obteniendo así tres imágenes de salida. Este parámetro puede aumentar debido al uso de filtros que necesitamos emplear en la imagen.

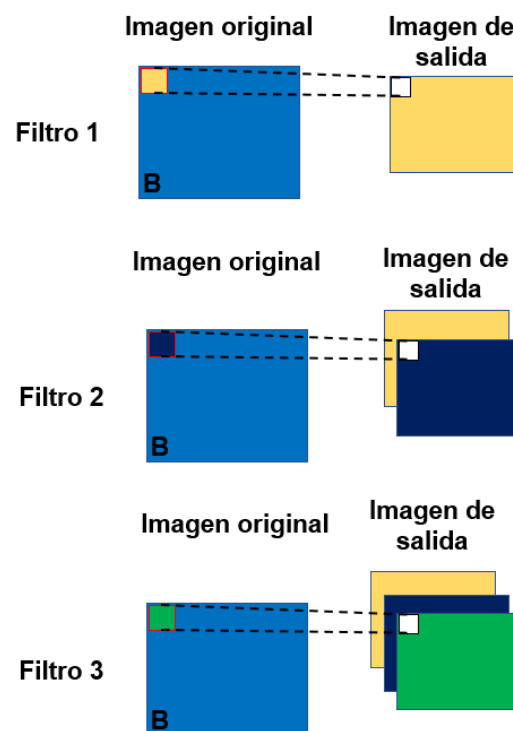


Imagen 55. Proceso de filtrado de la capa convolucional.

En cuanto al paso o stride del filtro, corresponde al salto que queremos que recorra el filtro a lo largo de la imagen que se está explorando. Puede realizarse mediante pixel a pixel o por grupo de pixeles. Por ejemplo, si el paso tiene como valor dos, el filtro se recorre cada dos pixeles (Imagen 56). Cuanto mayor es el paso más vamos a reducir el tamaño de la imagen de salida.

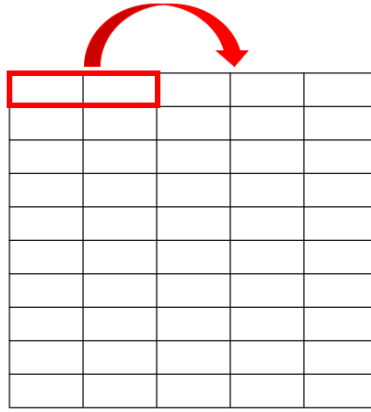


Imagen 56. Recorrido del filtro en la imagen de entrada de la capa convolucional.

- **Capa Max-Pooling o Average-Pooling:** Es la capa que realiza la reducción de los datos, es decir, reduce el ancho y el alto de las imágenes obtenidas de la capa convolucional, pero la profundidad la mantiene tal cual la recibió de la capa convolucional. Se basa en disminuir la cantidad de datos que presentan características comunes, para ello, realiza dos posibles tareas:
 - **Max-Pooling:** Calcula el máximo valor de una región determinada como se observa en la imagen 57.

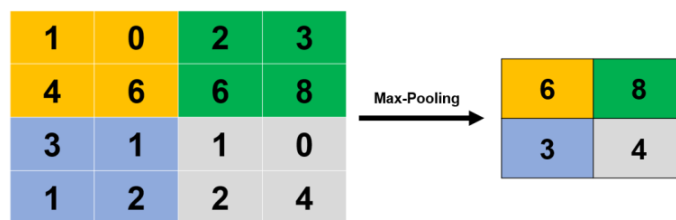


Imagen 57. Algoritmo Max-Pooling.

- **Average-Pooling:** Calcula el promedio de una región determinada como se observa en la imagen 58.

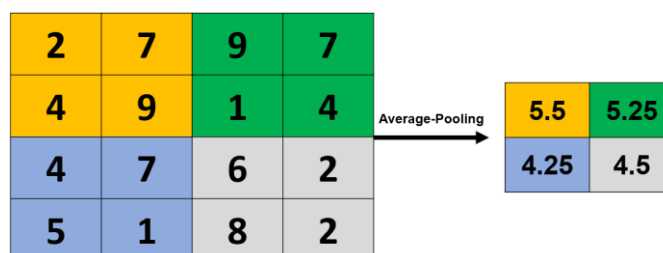


Imagen 58. Algoritmo Average-Pooling.

- **Capa clasificadora:** Capa que clasifica los datos obtenidos de la capa intermedia u oculta mediante clases predeterminadas.
- **Capa de salida:** Capa formada por neuronas encargadas de recibir los datos de la capa clasificadora para posteriormente tomar una decisión y ofrecer un resultado de salida.

Capítulo 6:

6. Método de detección y cuantificación de precipitación a partir de imágenes

En los capítulos anteriores hemos explicado el preprocesado de la señal recibida (capítulo 4) y el marco metodológico (capítulo 5). En este capítulo se expone la implementación del método que hemos utilizado para detectar y cuantificar el nivel de precipitación de lluvia a partir de imágenes recibidas desde diferentes cámaras de videovigilancia. Particularmente, para los experimentos utilizamos cámaras ubicadas en varios puntos de la isla de Gran Canaria.

6.1 Esquema general

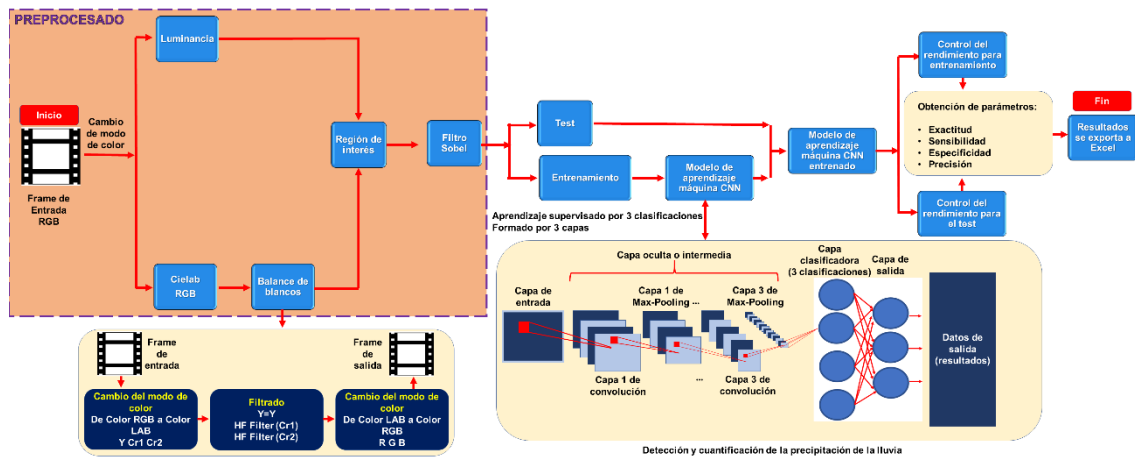


Imagen 59. Esquema general del TFM.

En el esquema general mostrado en la imagen 59 presentamos las técnicas aplicadas de aprendizaje máquina para la detección y cuantificación de lluvia sobre imágenes de videovigilancia de este TFM, con todos los elementos de la cadena hasta dar un resultado. A continuación, se procede a describir cada bloque del que está compuesto:

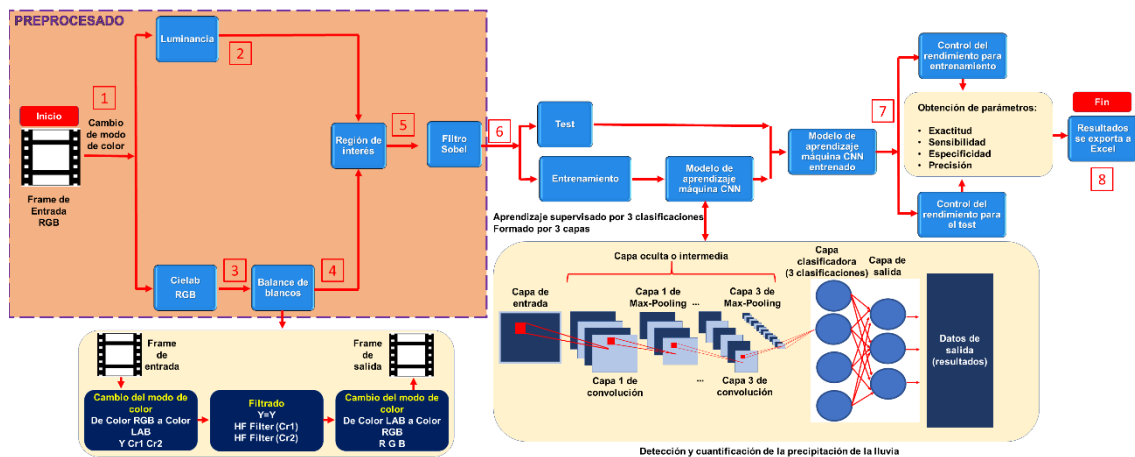


Imagen 60. Esquema general del TFM con una enumeración en cada paso.

Paso 1:

Inicialmente se recogen y se almacenan las imágenes procedentes de cámaras de videovigilancia. Estas imágenes están formadas por 25 imágenes o frames por segundo, en modo de color RGB (Imagen 61 y 62).



Imagen 61. Imagen de la cámara FOSCAM de día y sin lluvia.



Imagen 62. Imagen de la cámara FOSCAM de día y con lluvia.

Paso 2 y 3:

A estas imágenes se les va a realizar dos tipos de preprocesado. El primer paso del preprocesado es elegir el modo de color de la imagen. En el primer lugar,

pasaremos la imagen a modo de color luminancia (Imagen 63, 64 y 65) y, en segundo lugar, se hace un paso a Cielab RGB (Imagen 66, 67 y 68).



Imagen 63. Imagen de la cámara FOSCAM de día y sin lluvia con el modo de color luminancia.



Imagen 64. Imagen de la cámara FOSCAM de día y con lluvia con el modo de color luminancia.



Imagen 65. Esquema general del modo de color de la luminancia.



Imagen 66. Imagen de la cámara FOSCAM de día y sin lluvia con el modo de color cielab.



Imagen 67. Imagen de la cámara FOSCAM de día y con lluvia con el modo de color cielab.



Imagen 68. Esquema general del modo de color cielab.

Paso 4:

En el segundo procesado se realiza un balance de blancos, para normalizar los colores de las imágenes 69, 70 y 71.



Imagen 69. Imagen de la cámara FOSCAM de día y sin lluvia con el modo de color cielab después de aplicarle el filtro de balance de blancos.



Imagen 70. Imagen de la cámara FOSCAM de día y con lluvia con el modo de color cielab después de aplicarle el filtro de balance de blancos.



Imagen 71. Esquema general del modo de color cielab antes y después del filtro de balance de blancos.

Paso 5:

El siguiente paso en ambos preprocesados es seleccionar una región de interés.

Como se ha comentado en el capítulo 4, cada imagen está dividida en nueve posibles regiones de interés, como se ilustra en la siguiente ilustración 3:

| | | |
|----------|----------|----------|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Ilustración 3. Posibles regiones de interés en las que hemos dividido la imagen.

En las imágenes 72, 73 y 74 se observa la región de interés del modo de color luminancia y en las imágenes 75, 76 y 77 la del modo de color Cielab.



Imagen 72. Región de interés de la imagen de la cámara FOSCAM de día y sin lluvia con el modo de color luminancia.



Imagen 73. Región de interés de la imagen de la cámara FOSCAM de día y con lluvia con el modo de color luminancia.



Imagen 74. Esquema general del modo de color de la luminancia antes y después de la extracción de la región de interés.



Imagen 75. Región de interés de la imagen de la cámara FOSCAM de día y sin lluvia con el modo de color cielab.



Imagen 76. Región de interés de la imagen de la cámara FOSCAM de día y con lluvia con el modo de color cielab.



Imagen 77. Esquema general del modo de color de la cielab antes y después de la extracción de la región de interés.

Paso 6:

Una vez seleccionada la región de interés se realiza el filtrado Sobel para realzar los gradientes de la imagen.

En las imágenes 78, 79 y 80 se observa la región de interés del modo de color luminancia y en las imágenes 81, 82 y 83 la del modo de color Cielab.

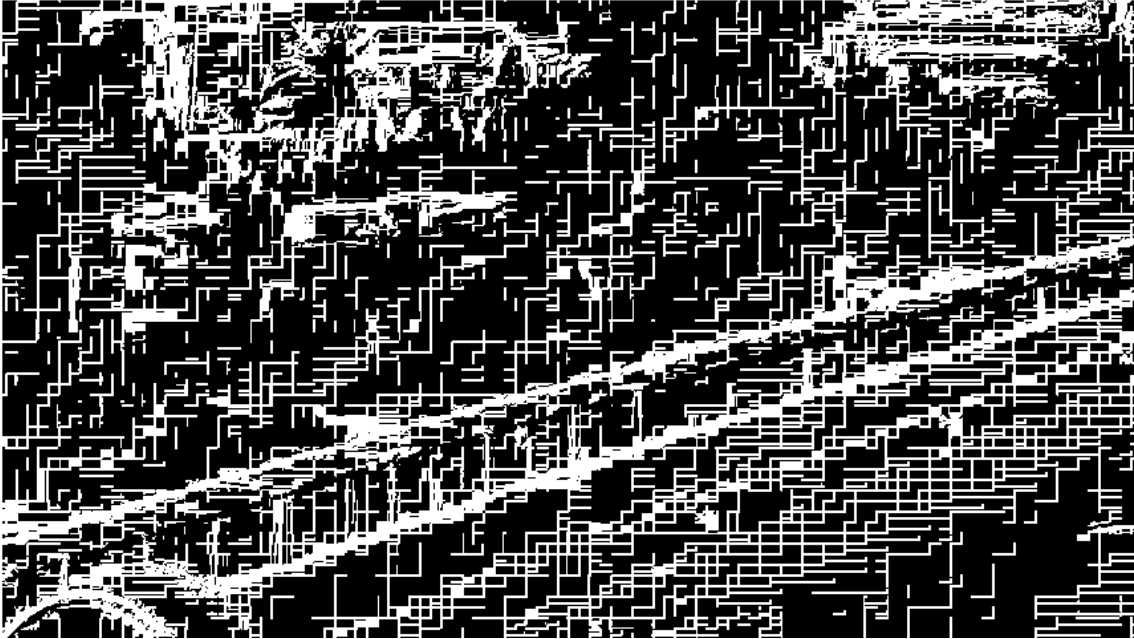


Imagen 78. Filtrado Sobel de la imagen de la cámara FOSCAM de día y sin lluvia con el modo de color luminancia.

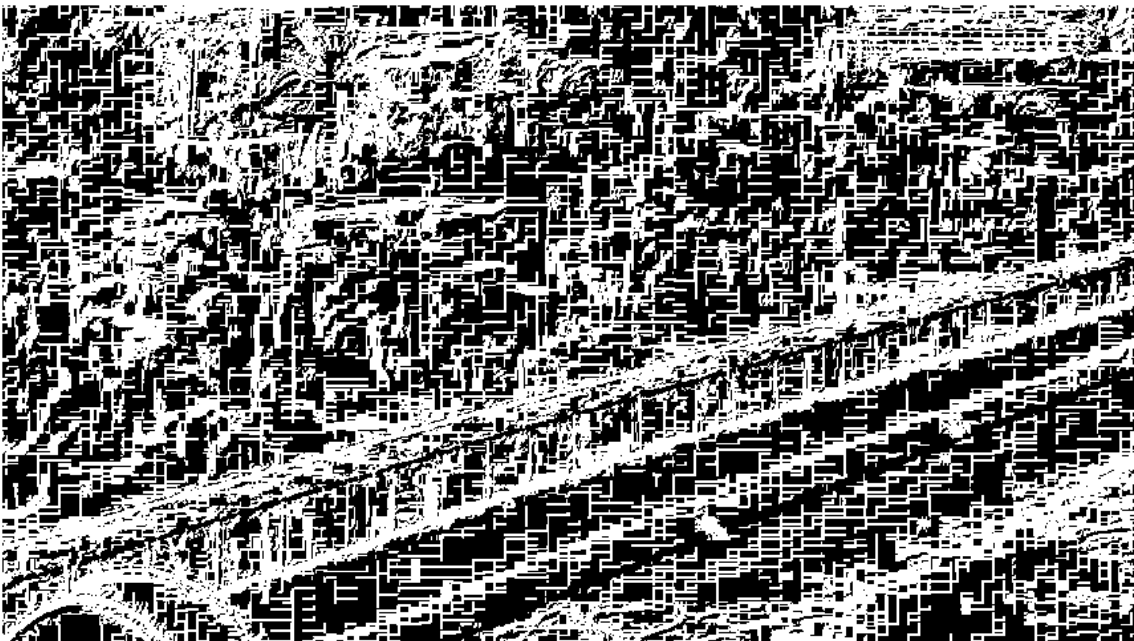


Imagen 79. Filtrado Sobel de la imagen de la cámara FOSCAM de día y con lluvia con el modo de color luminancia.

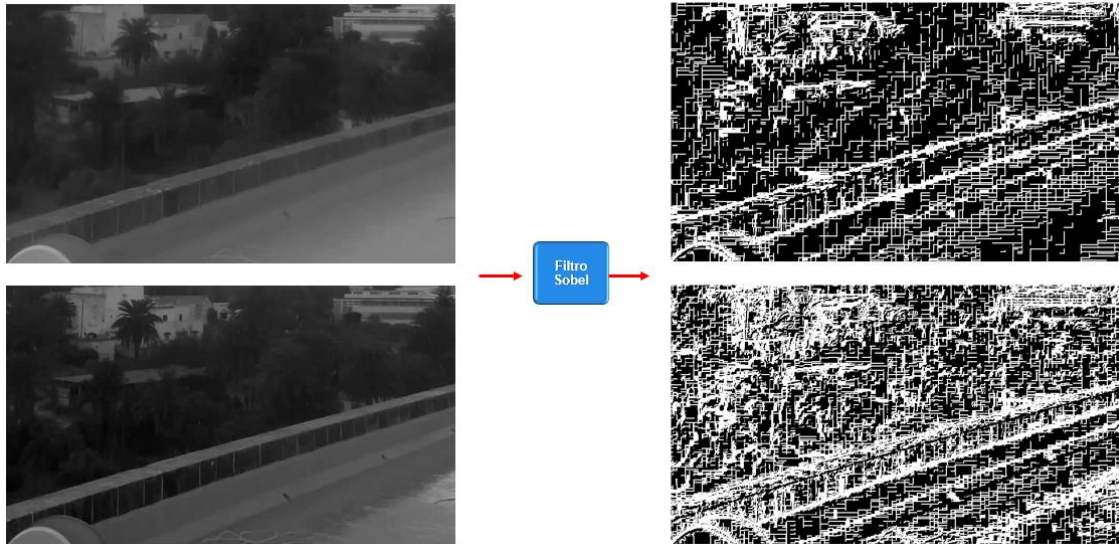


Imagen 80. Esquema general del modo de color de la luminancia antes y después del filtro Sobel.



Imagen 81. Filtrado Sobel de la imagen de la cámara FOSCAM de día y sin lluvia con el modo de color cielab.



Imagen 82. Filtrado Sobel de la imagen de la cámara FOSCAM de día y con lluvia con el modo de color cielab.

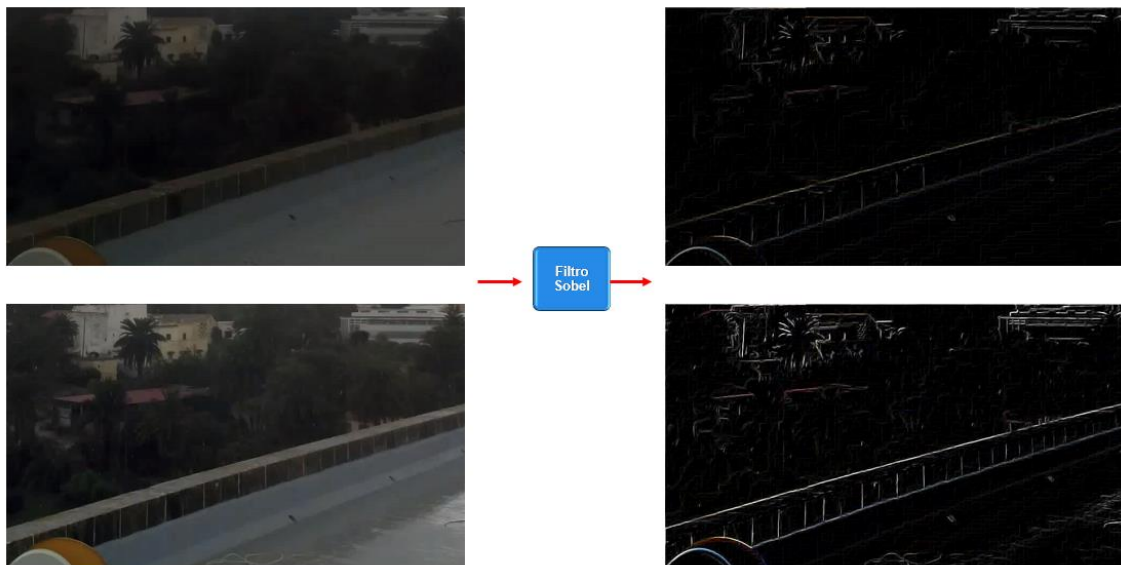


Imagen 83. Esquema general del modo de color cielab antes y después del filtro Sobel.

Paso 7:

Una vez realizada la parte del preprocesado de las imágenes procedemos a realizar la parte del entrenamiento y test de la red neuronal.

Los resultados obtenidos del modelo de aprendizaje máquina CNN entrenado se muestran en las imágenes 84 y 85.

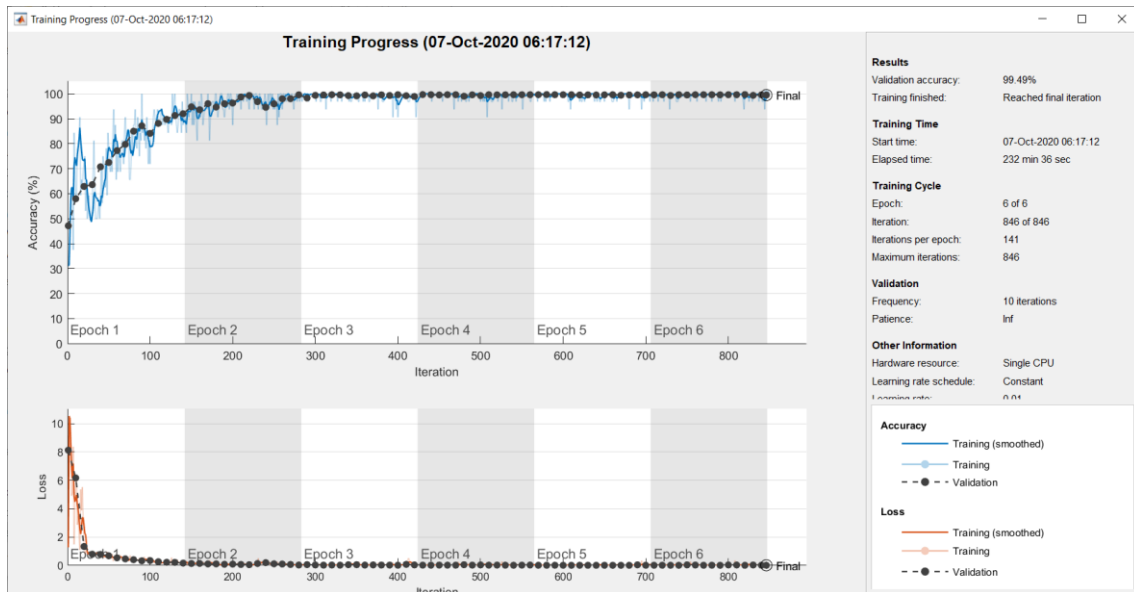


Imagen 84. Resultado del entrenamiento y test del modo de color cielab.

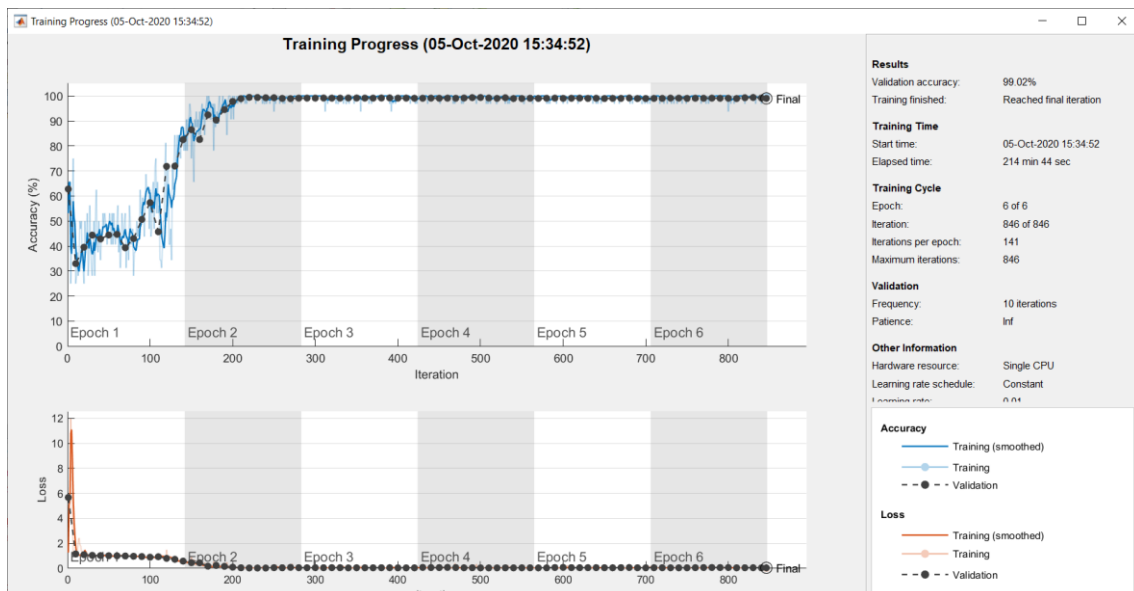


Imagen 85. Resultado del entrenamiento y test del modo de color luminancia.

Paso 8:

Los datos obtenidos se almacenan (Imagen 86) con las métricas de calidad, y que se explicarán con más detalle en el siguiente apartado.

Cada columna representa un parámetro de estudio

Cada fila representa una cámara de videovigilancia diferente

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,15 | 97,12 | 94,88 | 96,09 |
| 3 | 99,72 | 100,00 | 99,31 | 99,54 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 99,05 | 98,99 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 204 | 11 | 8 | 270 |
| 14 | 143 | 1 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 329 | 3 | 3 | 294 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 0 | 297 | 0 | 297 |

Imagen 86. Ejemplo de métricas de calidad almacenada.

Las filas del fichero Excel representa cada una de las cámaras de donde proceden las imágenes estudiadas:

- Fila 2: Tafira1-Informatica
- Fila 3: Tafira3-FOSCAM
- Fila 4: Hospital1-Barranquense
- Fila 5: Hospital2-Catedral
- Fila 6: Hospital3-Isleta
- Fila 7: Hospital4-Escaleritas
- Fila 8: Tafira2-MontanaArucas
- Fila 9: Tafira4-FOSCAM

Cada columna representa un parámetro de calidad de estudio:

- Exactitud
- Sensibilidad
- Especificidad
- Precisión

Para el modo de color Cielab se ilustra en la imagen 87 y para el modo de luminancia en la imagen 88.

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,15 | 97,12 | 94,88 | 96,09 |
| 3 | 99,72 | 100,00 | 99,31 | 99,54 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 99,05 | 98,99 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 204 | 11 | 8 | 270 |
| 14 | 143 | 1 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 329 | 3 | 3 | 294 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 0 | 297 | 0 | 297 |

Imagen 87. Tabla de los datos obtenidos con el modo de color luminancia.

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 56,39 | 100,00 | 0,00 | 56,39 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 97,14 | 94,17 | 100,00 | 100,00 |
| 6 | 79,65 | 56,90 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 93,94 | 87,88 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 0 | 215 | 0 | 278 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 232 | 0 | 13 | 210 |
| 17 | 332 | 0 | 128 | 169 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 297 | 0 | 36 | 261 |

Imagen 88. Tabla de los datos obtenidos con el modo de color cielab.

6.2 Métricas de calidad

Para estudiar la metodología diseñada se deben de definir unos parámetros que den idea de la calidad en la detección y la cuantificación. [Fuente 2020] [Manso 2020] [Kubat 2017].

6.2.1 Parámetros de calidad para la detección

Para definir los parámetros de calidad para la detección es necesario establecer, inicialmente, una matriz de confusión de 2x2. En ella, los parámetros usados son los siguientes:

- **TP:** en inglés “True Positive” o en castellano “Verdadero Positivo”. Se detecta lluvia y hay datos meteorológicos reales de lluvia.
- **FN:** en inglés “False Negative” o en castellano “Falso Positivo”. Se detecta no lluvia y hay datos meteorológicos reales de lluvia.
- **TN:** en inglés “True Negative” o en castellano “Verdadero Negativo”. Se detecta no lluvia y no hay datos meteorológicos reales de lluvia.
- **FP:** en inglés “False Positive” o en castellano “Falso Negativo”. Se detecta lluvia y no hay datos meteorológicos reales de lluvia.

Recordemos que disponemos de 2 clases en el aprendizaje supervisado de la red neuronal para el proceso de detección:

- **NL:** No hay datos meteorológicos reales de lluvia.
- **SL:** Si hay datos meteorológicos reales de lluvia.

En la tabla 6 se muestra los resultados obtenidos en la detección de la precipitación, se trata de una matriz de confusión binaria de tamaño 2x2. Las filas de dicha matriz representan la lluvia real u observada, mientras que las columnas representan la lluvia detectada o su predicción.

Tabla 6. Matriz de confusión binaria para la detección de la precipitación.

| | | Lluvia detectada o su predicción | |
|-------------------------|----|----------------------------------|----|
| | | NL | SL |
| Lluvia real u observada | NL | TN | FP |
| | SL | FN | TP |

En la tabla 7 se muestra la matriz de confusión anterior, pero con la interpretación de los resultados obtenidos en cada caso. Se dispone de cuatro posibles resultados:

Tabla 7. Matriz de confusión binaria con la interpretación de los resultados obtenidos en la detección de la precipitación.

| | | Lluvia detectada o predicción | |
|-------------------------|----|--|--|
| | | NL | SL |
| Lluvia real u observada | NL | No se detecta lluvia y no hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y no hay datos meteorológicos reales de lluvia. |
| | SL | No se detecta lluvia y si hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y si hay datos meteorológicos reales de lluvia. |

Para el estudio del rendimiento y la calidad de la detección de la precipitación se ha empleado los siguientes parámetros:

- **Exactitud:**

Este parámetro representa una proporción entre el número de eventos detectados como verdaderos positivos y verdaderos negativos entre todos los posibles resultados obtenidos. Este parámetro es importante, ya que, con él, podemos medir la frecuencia con la que el sistema realiza una predicción correcta. (Ecuación 4)

$$\text{Exactitud} = \frac{\text{n}^\circ \text{ de eventos detectados verdaderos positivos y negativos}}{\text{n}^\circ \text{ de eventos totales}} = \frac{TP+TN}{TP+FN+TN+FP}$$

Ecuación 4. Fórmula de la exactitud.

- **Sensibilidad:**

Este parámetro representa una proporción entre el número de eventos detectados como verdaderos positivos de lluvia entre el número de eventos totales reales u observados positivos de lluvia (hay datos meteorológicos reales de lluvia). (Ecuación 5)

$$\text{Sensibilidad} = \frac{\text{n}^\circ \text{ de eventos detectados positivos}}{\text{n}^\circ \text{ de eventos totales reales positivos}} = \frac{TP}{TP+FN}$$

Ecuación 5. Fórmula de la sensibilidad.

- **Especificidad:**

Este parámetro representa una proporción entre el número de eventos detectados como casos negativos de lluvia entre el número de eventos totales reales u observados negativos de lluvia (datos meteorológicos reales de no lluvia). (Ecuación 6)

$$\text{Especificidad} = \frac{\text{n}^\circ \text{ de eventos detectados negativos}}{\text{n}^\circ \text{ de eventos totales reales negativos}} = \frac{TN}{TN+FP}$$

Ecuación 6. Fórmula de la especificidad.

- **Precisión:**

Este parámetro representa una proporción entre el número de eventos detectados como casos positivos de lluvia entre el número de eventos totales reales u observados negativos de lluvia (datos meteorológicos reales de no lluvia). (Ecuación 7)

$$\text{Precisión} = \frac{\text{n}^\circ \text{ de eventos detectados positivos}}{\text{n}^\circ \text{ de eventos totales reales positivos}} = \frac{TP}{TP+FP}$$

Ecuación 7. Fórmula de la precisión.

6.2.2 Parámetros de calidad para la cuantificación

Para medir los parámetros de calidad para la cuantificación es necesario ampliar la matriz de confusión a un tamaño de 3x3.

En este momento disponemos de 3 clases en el aprendizaje supervisado de la red neuronal:

- **NL_NN:** No hay datos meteorológicos reales de lluvia y el nivel de precipitación es nulo.
- **SL_NB:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es bajo.
- **SL_NA:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es alto.

En la tabla 8 se muestra los resultados obtenidos en la detección de la precipitación, se trata de una matriz de confusión de tamaño 3x3. Las filas de dicha matriz representan la lluvia real u observada, mientras que las columnas representan la lluvia detectada o su predicción.

Tabla 8. Matriz de confusión para la detección de la precipitación.

| | | Lluvia detectada o su predicción | | |
|-------------------------|-------|----------------------------------|-------|-------|
| | | NL_NN | SL_NB | SL_NA |
| Lluvia real u observada | NL_NN | TN | FP | FP |
| | SL_NB | FN | TP | TP |
| | SL_NA | FN | TP | TP |

En la tabla 9 se muestra la matriz de confusión anterior, pero con la interpretación de los resultados obtenidos en cada caso. Se dispone de nueve posibles resultados:

Tabla 9. Matriz de confusión con la interpretación de los resultados obtenidos en la detección de la precipitación.

| | | Lluvia detectada o predicción | | |
|-------------------------|-------|--|--|--|
| | | NL_NN | SL_NB | SL_NA |
| Lluvia real u observada | NL_NN | No se detecta lluvia y no hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y no hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y no hay datos meteorológicos reales de lluvia. |
| | SL_NB | No se detecta lluvia y si hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y si hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y si hay datos meteorológicos reales de lluvia. |
| | SL_NA | No se detecta lluvia y si hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y si hay datos meteorológicos reales de lluvia. | Si se detecta lluvia y si hay datos meteorológicos reales de lluvia. |

Tal como se puede observar en la tabla 9, el sistema indica si se ha detectado lluvia o no y una cuantificación, y su grado de acierto es medible mediante los parámetros antes descritos. Si queremos aprovechar esos resultados para dar resultados de detección sólo, sin cuantificación, podemos hacer una distinción como la mostrada en la tabla 10. De esta forma podemos abordar el problema de detección y cuantificación simultáneamente.

Tabla 10. Matriz de confusión para la detección de la precipitación.

| | | Lluvia detectada o su predicción | | |
|-------------------------|-------|----------------------------------|-------|-------|
| | | NL_NN | SL_NB | SL_NA |
| Lluvia real u observada | NL_NN | TN | FP | FP |
| | SL_NB | FN | TP | TP |
| | SL_NA | FN | TP | TP |

Al usar los resultados de la forma mostrada en la tabla 10 podemos aplicar las métricas de calidad descritas en el apartado 1.2.1 (exactitud, precisión, sensibilidad y especificidad) [Kubat 2017] [Manso 2020] [Witten 2005].

Por esta razón necesitamos establecer un sistema que nos permita estudiar con mayor detalle la credibilidad de los resultados obtenidos.

6.2.3 Credibilidad de los parámetros de cuantificación

En la teoría de probabilidad y estadística, existe una distribución de tipo dicotómica denominada distribución de Bernoulli, es decir, una probabilidad discreta que contempla dos posibles resultados, éxito o fracaso. Mediante una variable aleatoria “X” se puede medir la probabilidad de éxito o fracaso de un experimento [Kubat 2017].

El clasificador funciona de la misma forma que esta distribución, ya que tras realizar un experimento tendremos éxito o fracaso en la detección de lluvia. La probabilidad de éxito (por ejemplo, la exactitud) se estima a partir de la suma de éxitos sobre el conjunto total de ejemplos. Tenemos pues una suma de variables aleatorias de tipo Bernoulli donde, por aplicación del Teorema Central del Límite, cuando el número de ejemplos (imágenes en nuestro caso) sobre los que se aplica la clasificación tiende a infinito la suma citada tiende a una variable aleatoria Gaussiana o Normal.

Ayudados de la suposición de gaussianidad, podemos estudiar el posible error del muestreo hecho por el clasificador, dado un nivel de confianza preestablecido. En estadística se estima y se estudia el citado error mediante el cálculo del intervalo de confianza. Este intervalo expresa un rango de valores donde se encuentra el resultado del experimento, obteniendo un resultado más fiable y preciso.

Por ejemplo, si buscamos un nivel de confianza del 95% (Tasa de éxito en la detección del 95%), esto indica que el 5% es el posible error cometido (imagen 89).

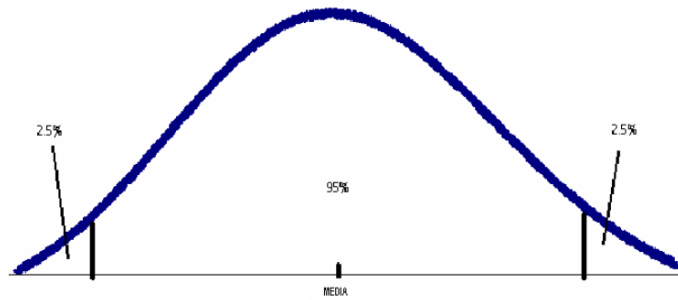


Imagen 89. Ejemplo de un intervalo de confianza al 95%.

Para poder realizar la aproximación de una distribución discreta tipo Bernulli a una distribución continua tipo Normal es necesario cumplir unas condiciones, conocidas como el teorema del límite central y vienen dadas en la ecuación 8.

En el caso de que una de estas condiciones no se cumpla no se podrá realizar dicha aproximación, se tendría que estudiar otro tipo de procedimiento.

Ecuación 8. Ecuaciones que debe de cumplir la suma de éxitos de la distribución de Bernulli para poder aproximarla a una distribución Normal.

$$p \cdot n \geq 10$$

$$q \cdot n \geq 10$$

Donde:

- **p**: probabilidad de éxito (Por ejemplo, exactitud)
- **q**: probabilidad de fracaso, donde $q=1-p$
- **n**: tamaño de la muestra, número de muestras que participan en el experimento. En nuestro caso el número de imágenes o fotogramas que usamos para el cálculo de la exactitud, sensibilidad, precisión y especificidad.

Según el Teorema de Moivre, la nueva variable aleatoria de la nueva distribución Normal para la proporción obtenida viene dada por (Ecuación 9):

Ecuación 9. Distribución normal obtenida para la proporción.

$$X' = N\left(p, \sqrt{\frac{pq}{n}}\right)$$

Para estandarizar se debe tipificar la variable aleatoria de la distribución Normal a una nueva variable Z (Ecuación 10):

Ecuación 10. Tipificación de la variable aleatoria.

$$Z = N(0,1) = \frac{X' - p}{\sqrt{\frac{pq}{n}}} = \frac{X' - \mu}{\sigma}$$

Donde:

- **Media** = $\mu = p$
- **Desviación típica** = $\sigma = \sqrt{\frac{pq}{n}}$

Como se observa en la ecuación 10, a la variable aleatoria de la distribución Normal se le resta la media y se divide entre la desviación típica.

Para calcular la probabilidad sería (Ecuación 11):

Ecuación 11. Probabilidad de la distribución Normal nueva obtenida.

$$\text{pr}[-k \leq Z \leq k] = \text{pr}\left[-k \leq \frac{X' - p}{\sqrt{\frac{pq}{n}}} \leq k\right] = c$$

El intervalo de confianza posee un ancho de $2k$.

Donde:

- La c es el nivel de confianza, luego $c = 1 - \alpha$. Esta variable no indica la probabilidad de que una variable aleatoria Z se encuentre dentro de un determinado rango de confianza.
- El nivel de significación es α .
- Valor crítico es $Z_{\alpha/2}$.

Por ejemplo, para obtener el valor crítico a partir de un nivel de confianza del 90% sería de la siguiente forma (Ecuación 12):

Ecuación 12. Valor crítico.

$$c = 1 - \alpha = 0,90$$

- Despejamos el nivel de significación (Ecuación 13):

Ecuación 13. Nivel de significación.

$$\alpha = 1 - 0,90 = 0,10$$

- Ahora dividimos entre dos el nivel de significación (Ecuación 14):

Ecuación 14. Dividimos el nivel de significación entre dos.

$$\frac{\alpha}{2} = \frac{0,10}{2} = 0,05 \rightarrow \text{es decir el 5\%}$$

- Le restamos uno a este valor (Ecuación 15):

Ecuación 15. Le restamos la unidad al valor obtenido en la ecuación anterior.

$$1 - \frac{\alpha}{2} = 1 - 0,05 = 0,95$$

- Por último, obtenemos el valor crítico en la tabla de N(0,1). Para ello, buscamos el valor obtenido en la ecuación 15 en el interior de la tabla y anotamos su valor sumando su valor en fila y columna, obteniendo así el valor crítico (ver, por ejemplo, la Tabla 11).

Tabla 11. Resultado de la tabla Gaussiana o tabla de la Normal.

| | |
|-----|--------|
| | 0,5 |
| 1,6 | 0,9505 |

El valor de $Z_{\alpha/2}$ es 1,645.

En el caso de que el valor que buscamos no se encuentre en la tabla de la Normal podemos realizar una interpolación lineal para obtenerlo.

El último paso necesario para estimar el intervalo de confianza es calcular la ecuación cuadrática con sus límites de confianza superior e inferior mostrada en la ecuación 16:

Ecuación 16. Ecuación cuadrática que da el intervalo de confianza.

$$p = \left(f + \frac{z^2}{2N} \pm z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) / \left(1 + \frac{z^2}{N} \right)$$

Donde:

- **p**: Parámetro de la calidad de la cuantificación, con este parámetro se obtiene el intervalo de confianza obtenido al aplicar \pm .
- **f**: valor de la exactitud, sensibilidad, precisión o especificidad obtenidos.
- **N**: tamaño de la muestra, número de muestras que participan en el experimento. En nuestro caso el número de imágenes o fotogramas que

usamos para el cálculo de la exactitud, sensibilidad, precisión y especificidad.

- **$Z=Z_{\alpha/2}$** : Valor crítico. Este valor nos va a indicar el ancho del rango de confianza.

Algunos ejemplos de valores críticos obtenidos de la ecuación 16 anterior son los mostrados en la siguiente tabla 12 [Witten 2005]:

Tabla 12. Ejemplo de parámetro de la calidad de la cuantificación con sus límites de confianza y sus correspondientes valores críticos [Witten 2005].

| Pr[X ≥ z] (%) | z |
|---------------|------|
| 0,1 | 3,09 |
| 0,5 | 2,58 |
| 1 | 2,33 |
| 5 | 1,65 |
| 10 | 1,28 |
| 20 | 0,84 |
| 40 | 0,25 |

Por lo tanto, a modo de resumen los pasos para calcular los límites de confianza son los siguientes:

1. Comprobar las condiciones que debe de cumplir la distribución de Bernulli para poder aproximarla a una distribución Normal (superposición de Normalidad) (ecuación 8).
2. Partiremos de un valor de confianza $c=1-\alpha$ y, a partir de él, calcularemos el valor crítico $Z_{\alpha/2}$.
3. Aplicamos la expresión cuadrática expresada en la ecuación 16. Esta ecuación nos aporta el intervalo o margen de confianza de nuestros valores de calidad de cuantificación. Y, mediante la observación del límite superior e inferior, al calcular la diferencia de ambos valores podemos verificar la credibilidad de los datos obtenidos. Cuanto menor sea el valor de esta diferencia mayor credibilidad poseerán los resultados obtenidos.

6.3 Criterios para realizar los experimentos

En este apartado se describirán los criterios establecidos para la realización de los experimentos. Los criterios establecidos son los siguientes:

- **Base de datos:** La creación y la organización de la base de datos para la detección y la cuantificación de la precipitación de la lluvia se detalla en el capítulo 3 de este TFM. Todas las imágenes recogidas son suministradas en condiciones reales por cámaras de videovigilancia del proyecto VIMETRIC [VIMETRI-MAC] desplegadas en la isla de Gran Canaria. Para la elaboración de la base de datos se ha tenido que estudiar dos bloques: Captación y Organización.
 - En la captación se ha estudiado los diferentes modelos de cámara, sus parámetros técnicos, ubicaciones, entre otros.
 - En la parte de organización nomenclatura de los vídeos almacenados, nombre y ubicaciones de la organización de las carpetas, entre otros.

Todos los vídeos de esta base de datos se han preprocesado previamente y posteriormente se ha entregado a la red neuronal convolucional (CNN), donde se han utilizado para el entrenamiento de la misma para la obtención de un modelo de aprendizaje máquina CNN.

- **Calidad de los datos:** La calidad de los datos recogidos se detalla en el capítulo 3 de este TFM. Este TFM se ha particularizado a recoger datos con un número de imágenes o fotogramas limitados con la siguiente calidad:
 - **Número de fotogramas/segundo:** 30 fotogramas/segundo.
 - **Duración de la secuencia o vídeo:** 2 segundos, aproximadamente.
 - **Ancho del fotograma:** 1280.
 - **Alto del fotograma:** 720.

Tras varios experimentos con diferentes números de fotogramas, se ha observado que, a partir de 30 fotogramas, los resultados obtenidos son muy satisfactorios y no se obtiene una mejora significativa a partir de dicho valor.

- **Extracción de características:** Hemos estudiado el interés de preprocesar las imágenes antes de aplicar la técnica de Deep Learning (en nuestro caso, red neuronal convolucional). Las imágenes de características resultantes se han utilizado para el entrenamiento y test de los modelos de aprendizaje máquina.

- **Red neuronal convolucional:** Los detalles de la red neuronal convolucional se detallan en el capítulo 5 de este TFM.

La CNN se basa en una serie de algoritmos que se engloban en dos grandes bloques: Test y Entrenamiento.

El objetivo es la obtención de un modelo de aprendizaje máquina CNN entrenado que sea capaz de detectar si hay lluvia o no, y en el caso de disponer de lluvia realizar una cuantificación de la precipitación.

Como se ha detallado en el capítulo 6 en el apartado de las métricas de calidad, este TFM se ha realizado mediante un aprendizaje supervisado formado por 3 clasificaciones:

- **NL_NN:** No hay datos meteorológicos reales de lluvia y el nivel de precipitación es nulo.
- **SL_NB:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es bajo.
- **SL_NA:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es alto.

En cuanto a las capas ocultas o intermedias se ha optado por el uso de 3 capas. Tras varios experimentos, se ha observado que el uso de 3 capas para el fin de este TFM alcanza unos resultados muy satisfactorios y no es necesario incrementar el número de capas.

- **Métricas de calidad:** Los parámetros de calidad para la detección y cuantificación de la precipitación de la lluvia están recogidos anteriormente en este capítulo 6 de este TFM. En este TFM se ha realizado mediante un aprendizaje supervisado donde participan 2 (NL, SL) o 3 clases (NL_NN, SL_NB, SL_NA), según proceda.

Los parámetros de calidad se estiman en tres grupos de estudio:

- **Detección simple:** Tenemos dos clases, NL y SL, estimamos los parámetros de: exactitud, precisión, sensibilidad y especificidad, así como los intervalos de confianza.
- **Cuantificación-detección:** Para el estudio de la calidad para la detección establecemos una matriz de confusión de 3x3, tal como ya se explicó. Una vez establecida dicha matriz se procede a estudiar el rendimiento e interpretar la calidad de la detección de la precipitación mediante los parámetros de: exactitud, precisión, sensibilidad y especificidad, así como los intervalos de confianza.

- **Cuantificación:** Para el estudio de la calidad para la cuantificación establecemos una matriz de confusión de 3x3, al igual que en el caso anterior. Una vez establecida dicha matriz se procede a estudiar el rendimiento e interpretar la calidad de la detección de la precipitación mediante los parámetros de: exactitud, precisión, sensibilidad y especificidad, así como los intervalos de confianza.

6.4 Experimentos de detección y cuantificación. Resultados

El desarrollo y experimentación de este TFM se realizó en dos bloques de estudio diferentes:

- **El bloque de la detección:** En este bloque el objetivo es obtener un clasificador binario, que detecta si hay precipitación de lluvia real o no. En este mismo bloque realizamos un estudio de pre-procesado o pre-extracción de características que nos ayudara a mejorar la calidad de los datos obtenidos y así mejorar los resultados finales. Posteriormente se calcularon los parámetros de calidad para la detección (Exactitud, precisión, sensibilidad y especificidad) y su nivel de credibilidad.
- **El bloque de la cuantificación:** En este bloque ampliamos la matriz de confusión del bloque de detección y añadimos 3 clases al clasificador de la red neuronal. Ahora no sólo se detectaba la presencia o ausencia de lluvia, sino que además se clasificaba en tres niveles de lluvia (no lluvia, lluvia baja y lluvia alta). Y posteriormente se calcularon los parámetros de calidad para la cuantificación y su nivel de credibilidad.

En este TFM se ha optado por el uso de una red neuronal convolucional profunda de 3 capas intermedias u ocultas y un aprendizaje supervisado. Para la elección del número de capas, inicialmente hemos realizado un preprocesado, donde realizamos experimentos tales como balances de blancos, extracción de regiones de interés y añadimos filtros de luminancia, Cielab y Sobel. Una vez estudiado y seleccionado el bloque de preprocesado, se procedió a estudiar el número de capas mínimas necesarias en la red neuronal convolucional para la obtención de unos resultados óptimos.

Realizamos todos los experimentos teniendo en cuenta:

- Imágenes con lluvia y sin lluvia.
- Diferentes cámaras de videovigilancia.
- Diferentes momentos del día (noche y día).

6.4.1 Experimentos con Luminancia

Inicialmente se capturan una serie de imágenes o fotogramas desde las cámaras de videovigilancia y se van almacenando en una base de datos. Posteriormente, con cada una de esas imágenes o fotogramas se realizó un experimento de cambio de color. En este experimento se usó el modo de color de luminancia (blanco y negro) (Imágenes 90 y 91).

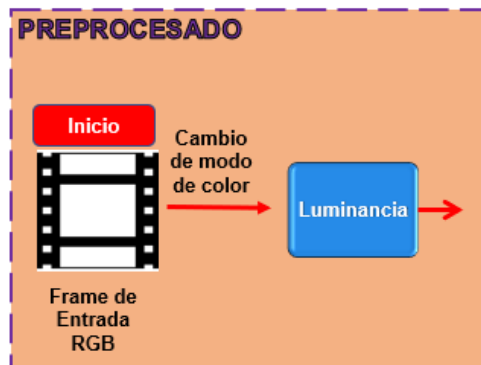


Imagen 90. Filtrado de modo de color a luminancia.



Imagen 91. Imágenes antes y después del filtrado de luminancia.

En la imagen 92 se observa, en la entrada del filtrado de luminancia, una imagen o fotograma de la cámara FOSCAM 1, ubicada en Tafira. Esta imagen está

tomada durante la noche y con un nivel de cuantificación de lluvia alta. El cambio de modo de color es poco apreciable.

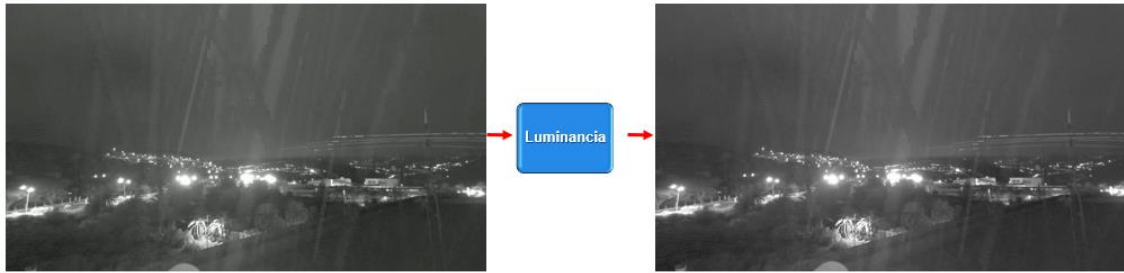


Imagen 92. Imagen de la cámara FOSCAM durante la noche y con lluvia alta.

En la imagen 93 se observa, en la entrada del filtrado de luminancia, una imagen o fotograma de la cámara Hospital 3-La Isleta, ubicada en el Hospital Rey Juan Carlos I. Esta imagen está tomada durante el día y con un nivel de cuantificación de lluvia alta. El cambio de modo de color es apreciable.



Imagen 93. Imagen de la cámara Hospital 3- La Isleta durante el día y con lluvia alta.

Los resultados obtenidos en el bloque de detección habían mejorado con el uso de este filtrado con las imágenes que eran de día, las de la noche no se apreciaba un cambio en los resultados significativos.

6.4.2 Experimentos con Cielab

Al igual que en el experimento anterior, inicialmente se captura una serie de imágenes o fotogramas desde las cámaras de videovigilancia y se van almacenando en una base de datos. Posteriormente, con cada una de esas imágenes o fotogramas se realizó un experimento de cambio de color. En este experimento se usó el modo de color de Cielab (Imágenes 94 y 95).

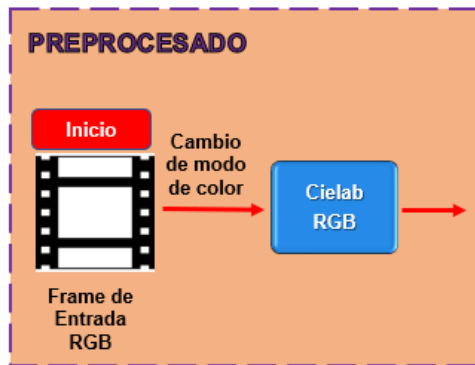


Imagen 94. Filtrado de modo de color a Cielab.



Imagen 95. Imágenes antes y después del filtrado de Cielab.

Como se ha comentado anteriormente en capítulos anteriores, este modo de color contiene un espacio de color tridimensional que describe todos los colores que puede distinguir el ojo humano. Se descompone de 3 componentes:

- **Componente L:** luminosidad, variación de negro a blanco.
- **Componente A:** croma, variación de rojo a verde.
- **Componente B:** croma, variación de azul a amarillo.

Las cámaras utilizadas son de alta calidad y están bien balanceadas. Luego el cambio es poco apreciable en las imágenes de salida del filtrado de Cielab.

En la imagen 96 se observa, en la entrada del filtrado de Cielab, una imagen o fotograma de la cámara FOSCAM 1, ubicada en Tafira. Esta imagen está tomada durante la noche y con un nivel de cuantificación de lluvia alta. El cambio de modo de color es poco apreciable.

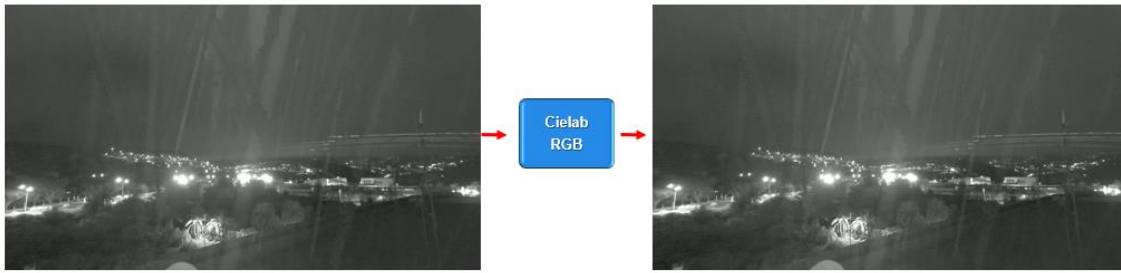


Imagen 96. Imagen de la cámara FOSCAM durante la noche y con lluvia alta.

En la imagen 97 se observa, en la entrada del filtrado de Cielab, una imagen o fotograma de la cámara Hospital 3-La Isleta, ubicada en el Hospital Rey Juan Carlos I. Esta imagen está tomada durante el día y con un nivel de cuantificación de lluvia alta. El cambio de modo de color es poco apreciable.



Imagen 97. Imagen de la cámara Hospital 3- La Isleta durante el día y con lluvia alta.

Los resultados obtenidos en el bloque de detección habían mejorado un poco con el uso de este filtrado.

6.4.3 Experimentos con balance de blancos

Este experimento se realizó a continuación del modo de color Cielab, ya que con el modo de color de luminancia el efecto de este filtro es despreciable.

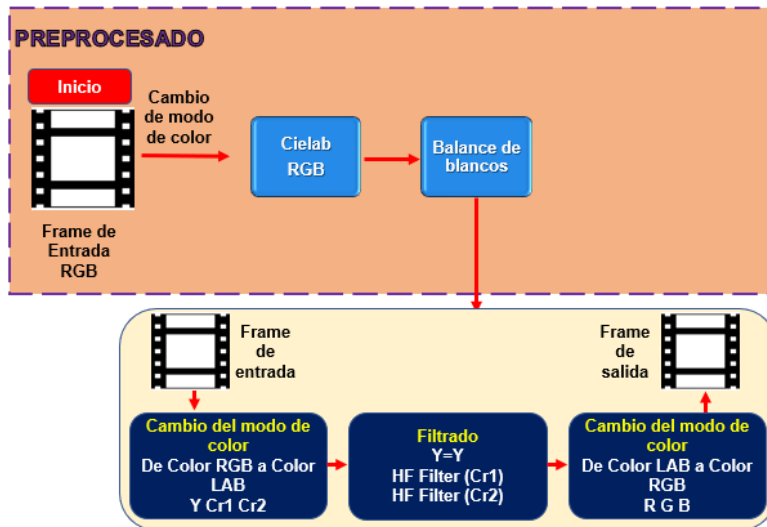


Imagen 98. Balance de blancos.



Imagen 99. Imágenes antes y después del filtrado de Balance de Blancos.

En la imagen 100 se observa, en la entrada del filtrado de Balance de Blancos, una imagen o fotograma de la cámara FOSCAM 1, ubicada en Tafira. Esta imagen está tomada durante la noche y con un nivel de cuantificación de lluvia alta. El resultado es poco apreciable.

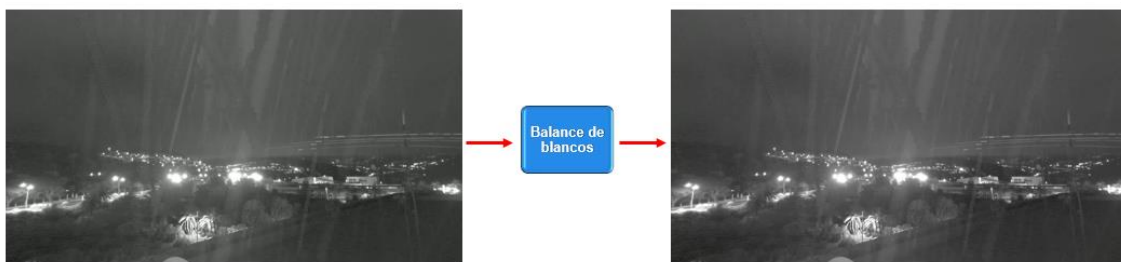


Imagen 100. Imagen de la cámara FOSCAM durante la noche y con lluvia alta.

En la imagen 101 se observa, en la entrada del filtrado de Balance de Blancos, una imagen o fotograma de la cámara Hospital 3-La Isleta, ubicada en el Hospital Rey Juan Carlos I. Esta imagen está tomada durante el día y con un nivel de cuantificación de lluvia alta. El resultado es apreciable.



Imagen 101. Imagen de la cámara Hospital 3- La Isleta durante el día y con lluvia alta.

Los resultados obtenidos en el bloque de detección habían mejorado con el uso de este filtrado.

6.4.4 Experimentos con Sobel

Una vez realizado con cambios de modo de color se introducen las imágenes en el filtro de Sobel (Imagen 102).

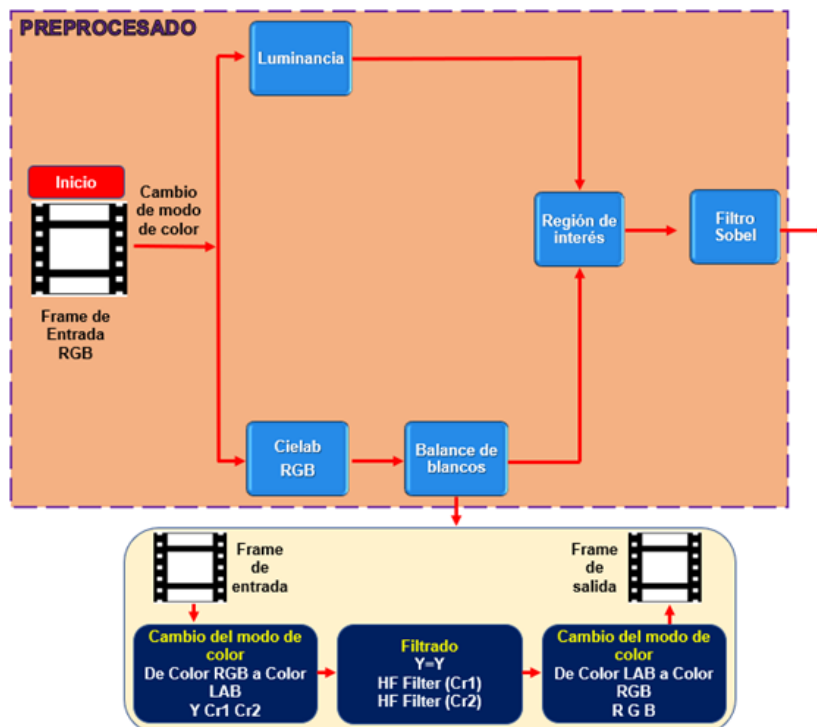


Imagen 102. Filtro Sobel.

En la imagen 103 se observa la entrada de dos imágenes, una de ellas tiene modo de color de luminancia y la otra imagen tiene un modo de color Cielab. Ambas imágenes son sin lluvia.

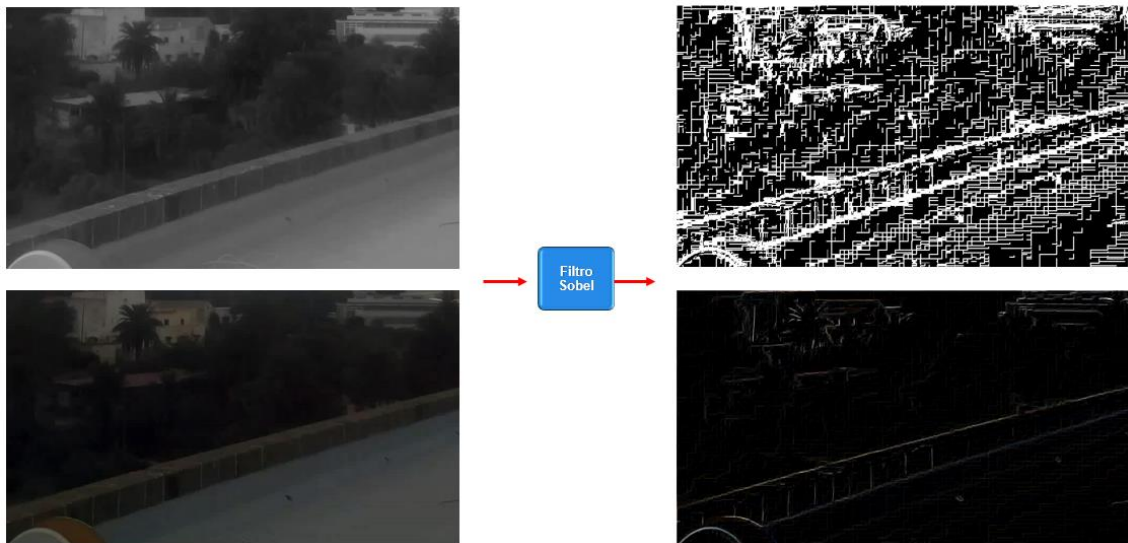


Imagen 103. Imagen de luminancia y Cielab, antes y después, del filtrado de Sobel sin lluvia.

En la imagen 104 se observa la entrada de dos imágenes, una de ellas tiene modo de color de luminancia y la otra imagen tiene un modo de color Cielab. Ambas imágenes son con lluvia.



Imagen 104. Imagen de luminancia y Cielab, antes y después, del filtrado de Sobel con lluvia.

En la imagen 105 se muestra los resultados de entrenamiento de la red obtenidos con una de las cámaras (Hospital1-Barrancoseco) sin el uso del filtro Sobel con el modo de color Cielab.

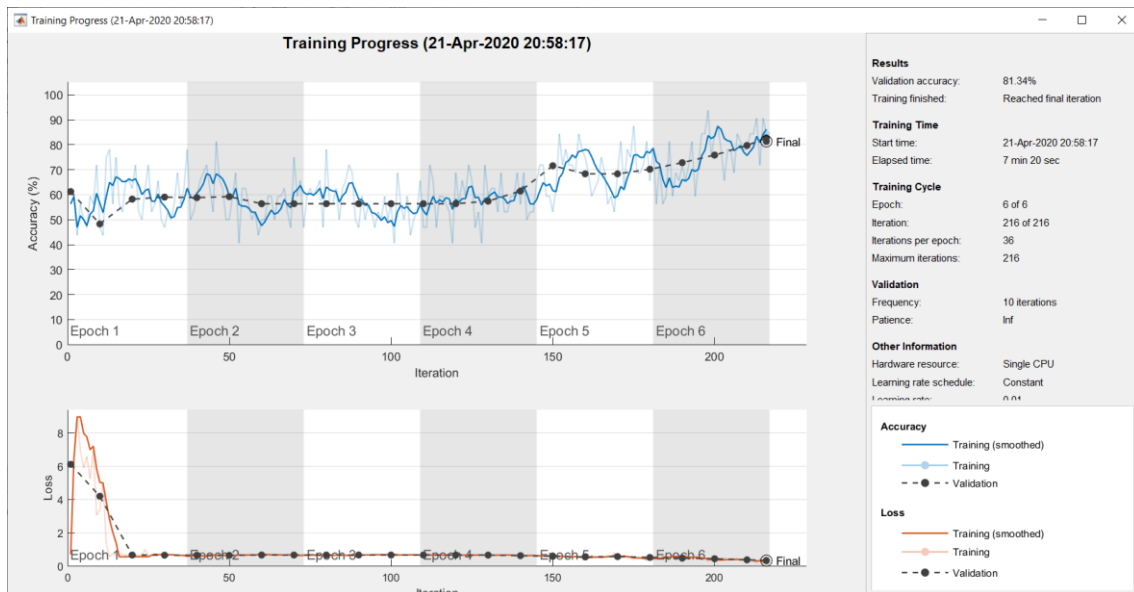


Imagen 105. Resultados obtenidos sin el uso del filtro Sobel con el modo de color de Cielab.

En la imagen 106 se muestra los resultados de entrenamiento de la red obtenidos con la misma cámara con el uso del filtro Sobel con el modo de color Cielab.

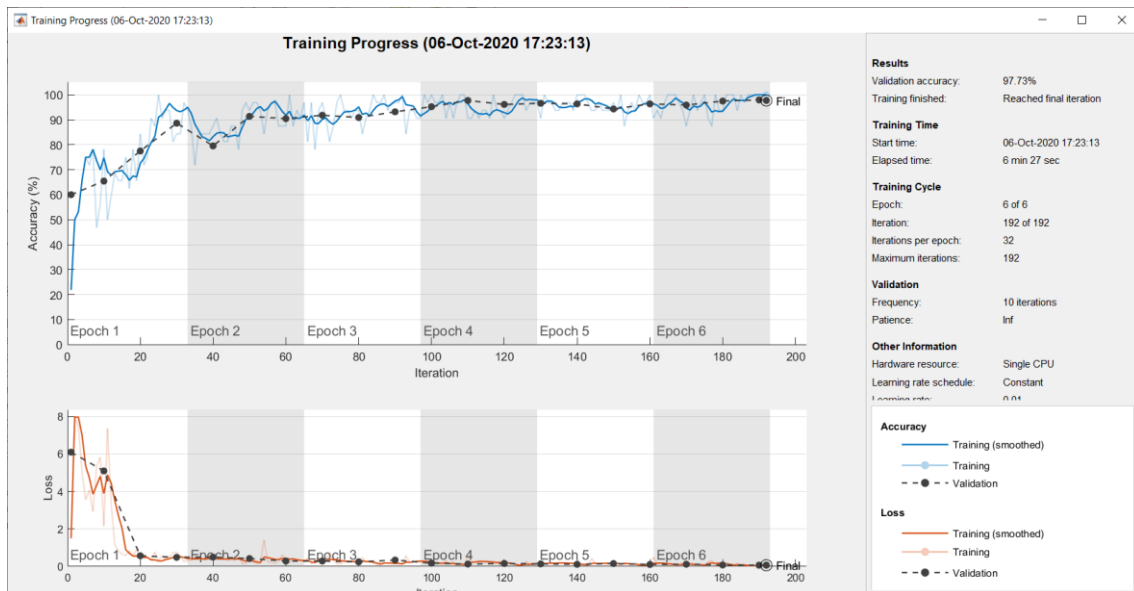


Imagen 106. Resultados obtenidos con el uso del filtro Sobel con el modo de color de Cielab.

En la imagen 107 se muestra los resultados de entrenamiento de la red obtenidos con una de las cámaras (Hospital1-Barrancoseco) sin el uso del filtro Sobel con el modo de color Luminancia.

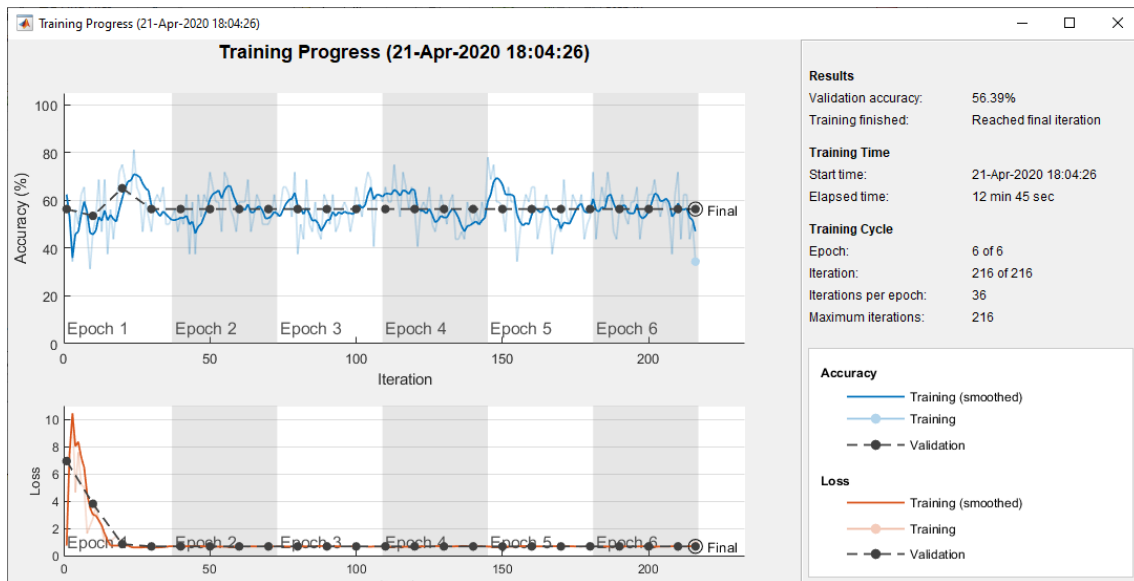


Imagen 107. Resultados obtenidos sin el uso del filtro Sobel con el modo de color de Luminancia.

En la imagen 108 se muestra los resultados de entrenamiento de la red obtenidos con la misma cámara con el uso del filtro Sobel con el modo de color Luminancia.

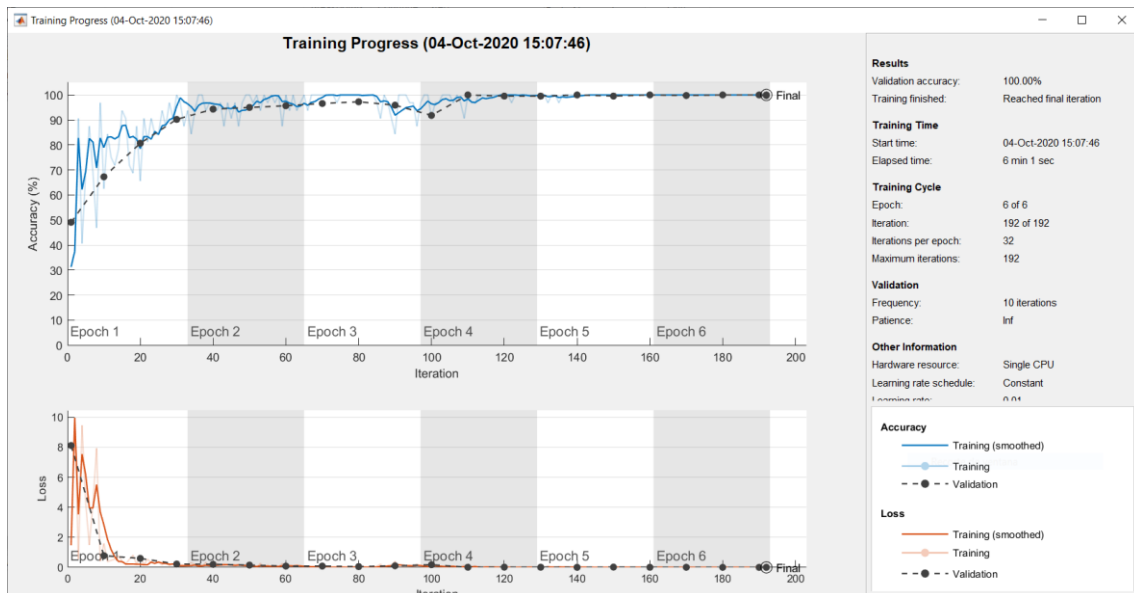


Imagen 108. Resultados obtenidos con el uso del filtro Sobel con el modo de color de Luminancia.

Como se observa en las imágenes anteriores los resultados obtenidos en el bloque de detección habían mejorado con el uso de este filtrado.

6.4.5 Experimentos sobre la profundidad de la red neuronal.

Una vez finalizada la etapa de preprocesado se procede a realizar los experimentos con la red neuronal convolucional.

Inicialmente se realizaron experimentos para el proceso de la detección de la precipitación de la lluvia mediante dos únicas clases en el aprendizaje supervisado de la red neuronal convolucional. Recordemos que estas dos clases son:

- **NL:** No hay datos meteorológicos reales de lluvia.
- **SL:** Si hay datos meteorológicos reales de lluvia.

Creamos una matriz de confusión binaria de tamaño 2x2 como se ha descrito en los capítulos anteriores.

Realizamos experimentos con 1, 2 y 3 capas intermedias u ocultas en la red. Y observamos que, a partir de la tercera capa, los resultados no mejoraban significativamente. Por esta razón sólo añadimos 3 capas intermedias.

El aprendizaje supervisado que hemos realizados consta de dos etapas, una de test y otra de entrenamiento. A la red neuronal se le aporta una base de datos donde se dispone de las dos clasificaciones de las imágenes antes descritas y, con ellas, comienza el proceso de aprendizaje máquina.

El diagrama de bloque del sistema de detección es el ilustrado en la imagen 109.

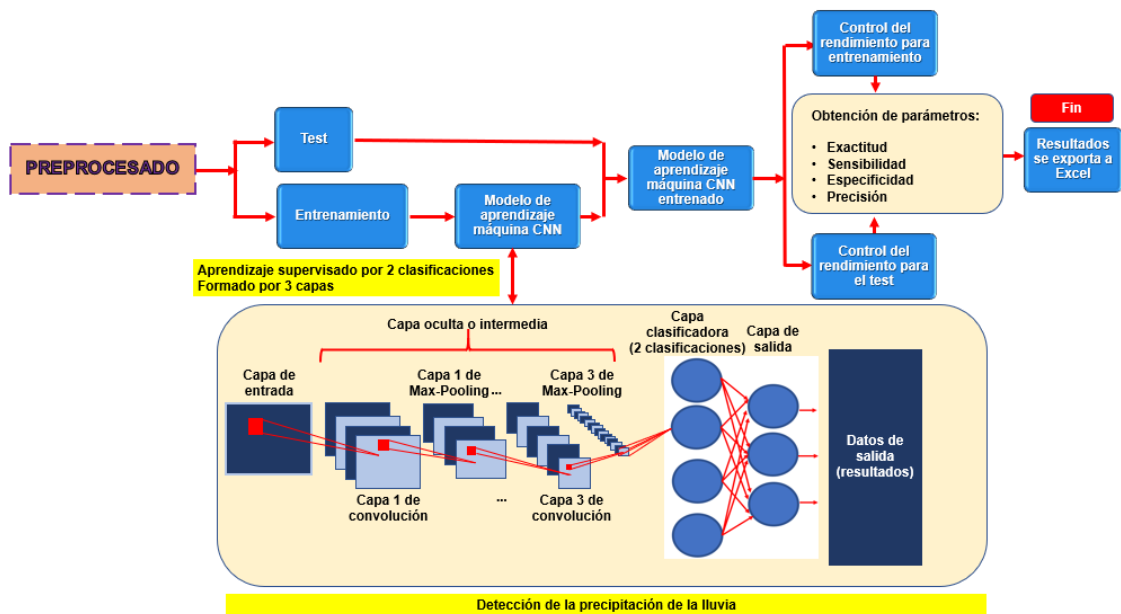


Imagen 109. Diagrama de bloques del sistema de detección incluyendo la etapa de la red neuronal convolucional.

Los resultados obtenidos son los siguientes:

- Resultados usando el modo de color luminancia, toda la etapa de procesado descrita con anterioridad y mediante un aprendizaje supervisado por 2 clasificaciones y 3 capas intermedias. Se realizaron 9 iteraciones, en la Tabla 13 se muestra una iteración, pero se adjuntan todas las iteraciones obtenidas en el Anexo 2.

Tabla 13. Resultados de la primera iteración con el modo de color luminancia en todas las cámaras.

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,15 | 97,12 | 94,88 | 96,09 |
| 3 | 99,72 | 100,00 | 99,31 | 99,54 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 99,05 | 98,99 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 204 | 11 | 8 | 270 |
| 14 | 143 | 1 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 329 | 3 | 3 | 294 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 0 | 297 | 0 | 297 |

A continuación, se muestra las gráficas de la primera iteración para todas las cámaras.

- **Primera iteración:**

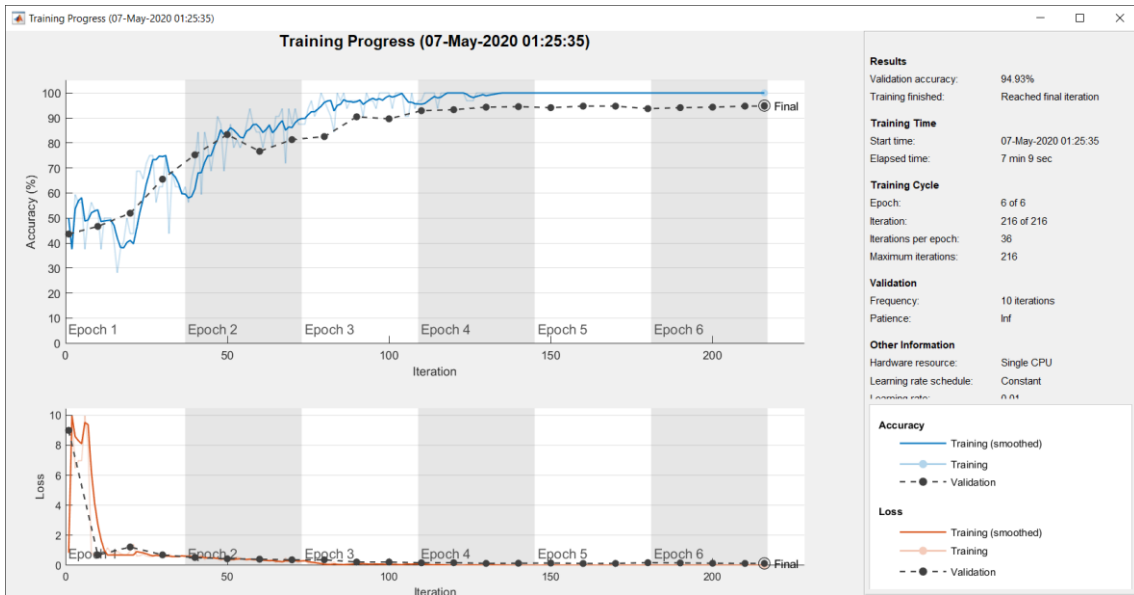


Imagen 110. Cámara Hospital1-Barranoseco.

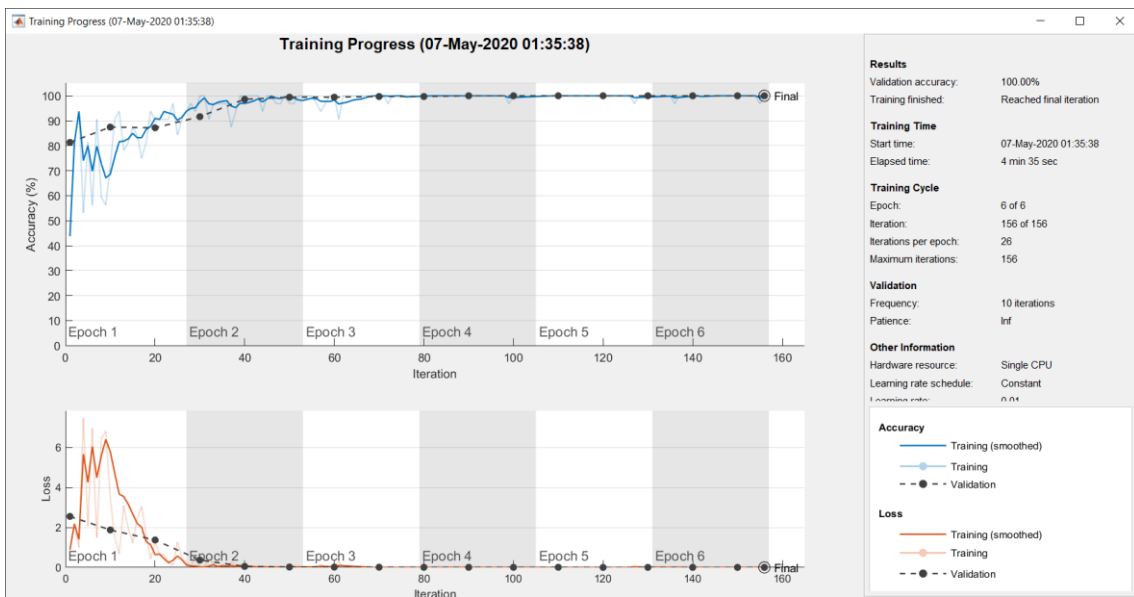


Imagen 111. Cámara Hospital2-Catedral.

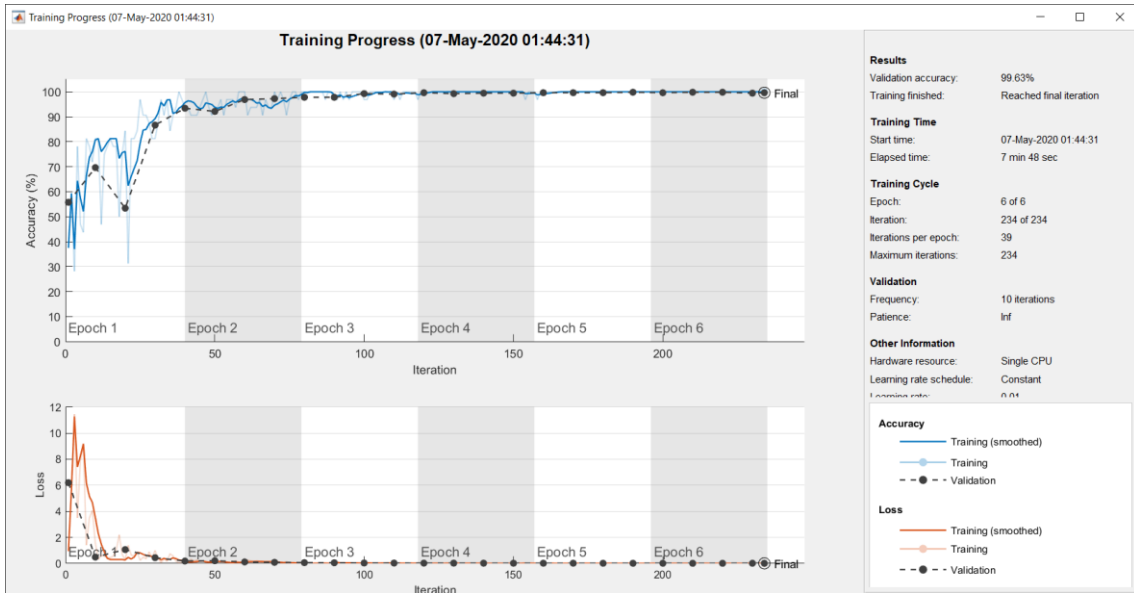


Imagen 112. Cámara Hospital3-Isleta.

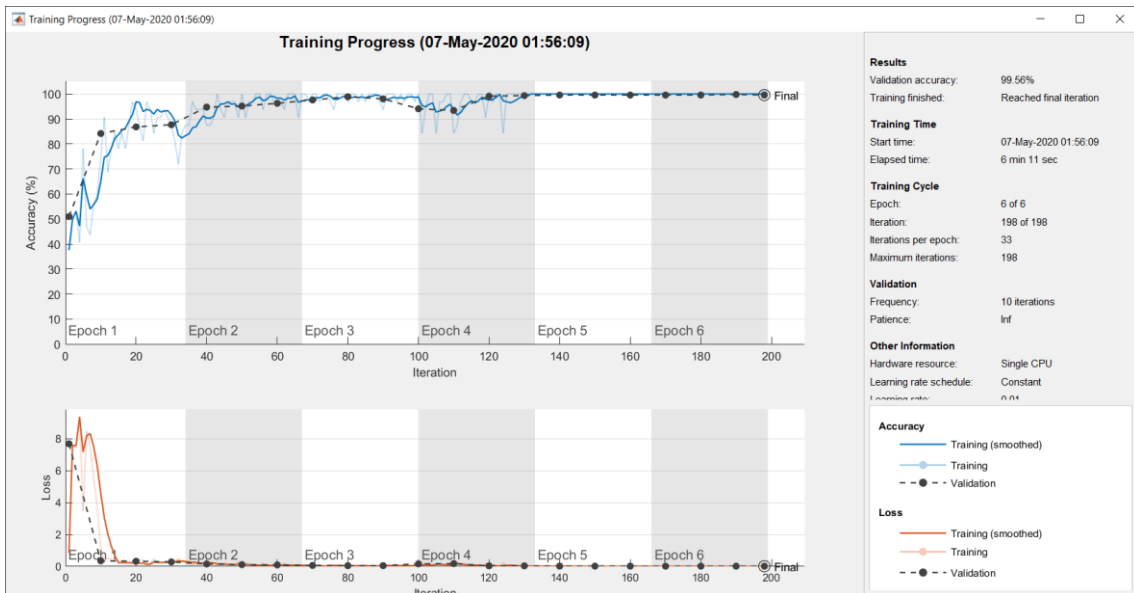


Imagen 113. Cámara Hospital4-Escaleritas.

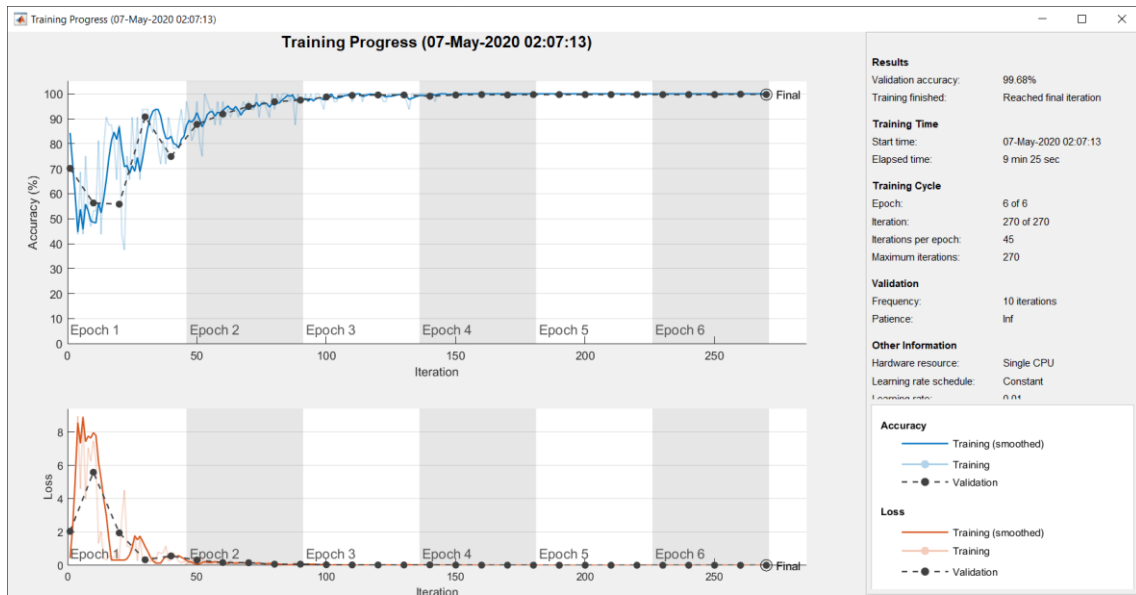


Imagen 114. Cámara Tafira1-Infornática.

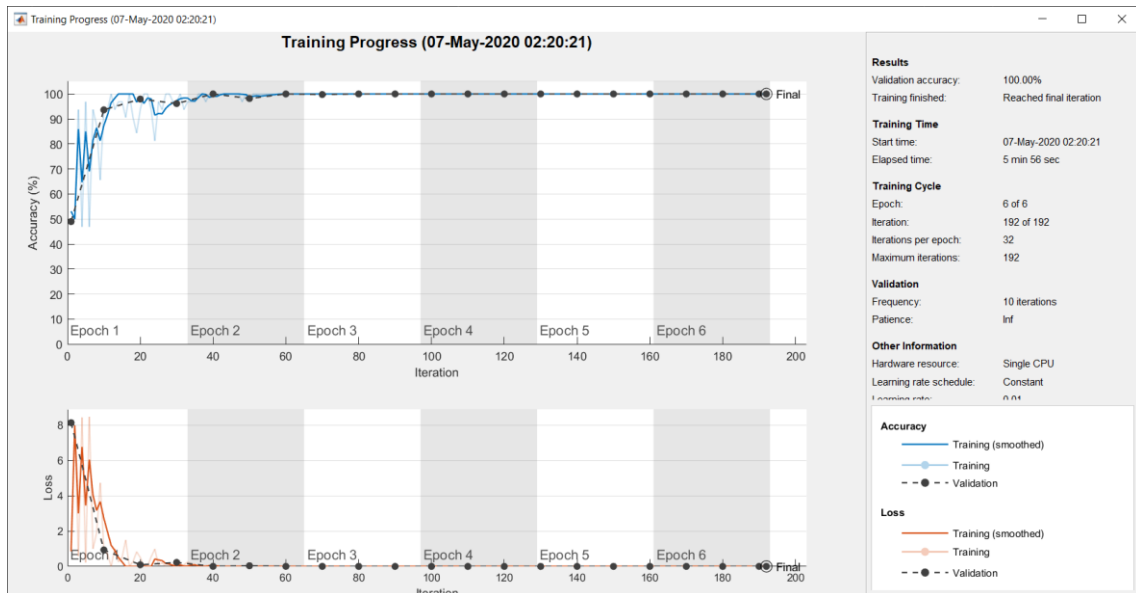


Imagen 115. Cámara Tafira2-MontañaArucas.

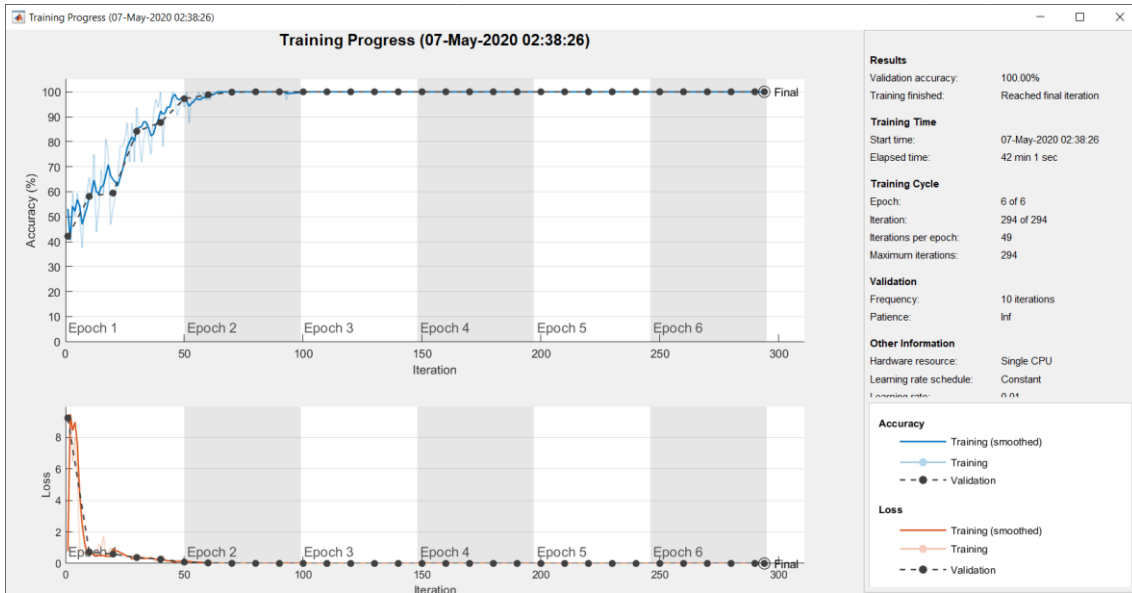


Imagen 116. Cámara Tafira3-Foscam.

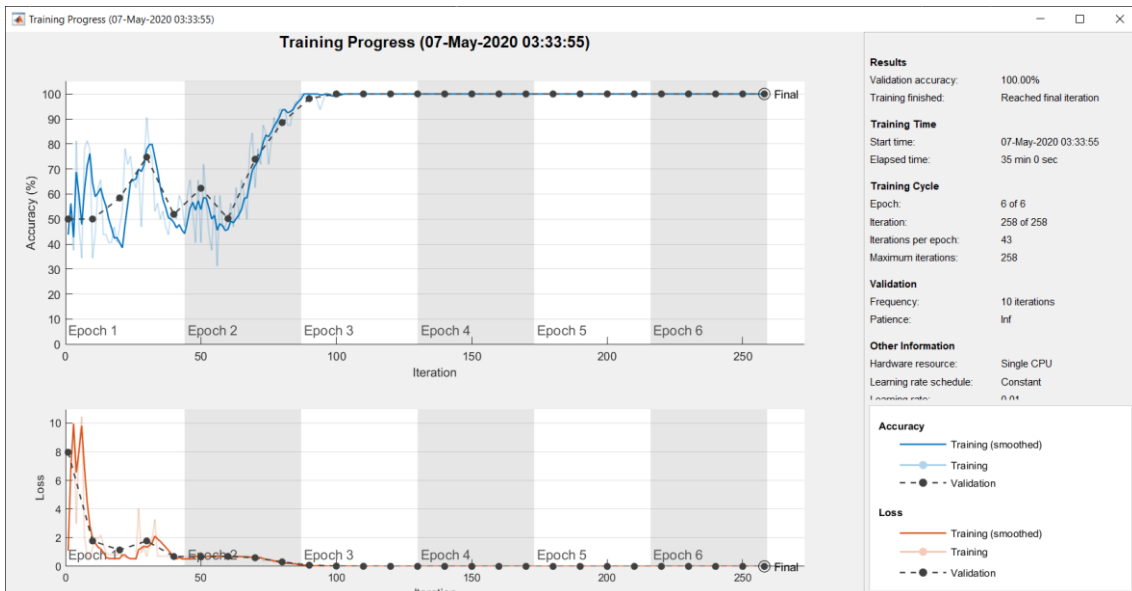


Imagen 117. Cámara Tafira4-Foscam.

- Resultados usando el modo de color Cielab, toda la etapa de procesamiento descrita con anterioridad y mediante un aprendizaje supervisado por 2 clasificaciones y 3 capas intermedias. Se han realizado 9 iteraciones. La primera iteración se muestra en la Tabla 14, las nueve iteraciones se pueden ver en el Anexo 2.

Tabla 14. Resultados de la primera iteración con el modo de color Cielab en todas las cámaras.

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 56,39 | 100,00 | 0,00 | 56,39 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 97,14 | 94,17 | 100,00 | 100,00 |
| 6 | 79,65 | 56,90 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 93,94 | 87,88 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 0 | 215 | 0 | 278 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 232 | 0 | 13 | 210 |
| 17 | 332 | 0 | 128 | 169 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 297 | 0 | 36 | 261 |

A continuación, se muestra las gráficas de la primera iteración para todas las cámaras.

- **Primera iteración:**

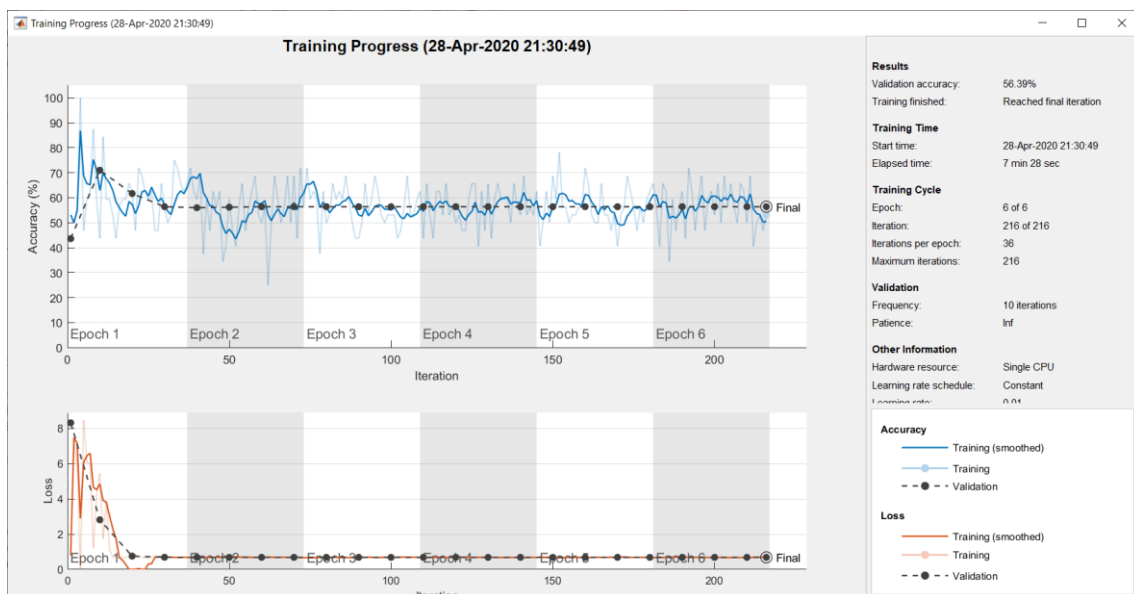


Imagen 118. Cámara Hospital1-Barrancoseco.

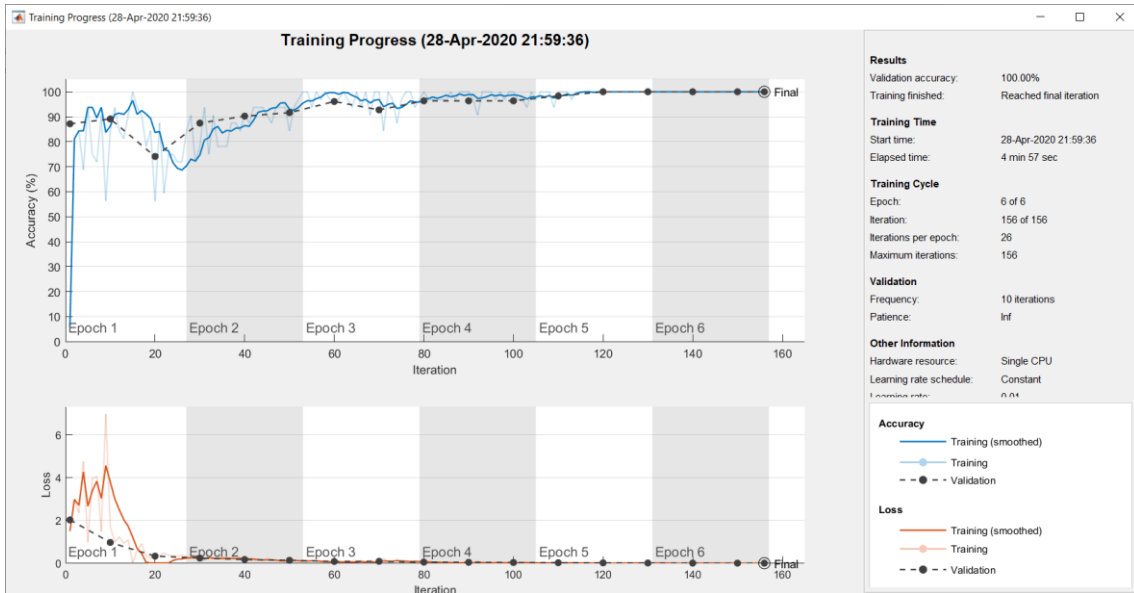


Imagen 119. Cámara Hospital2-Catedral.

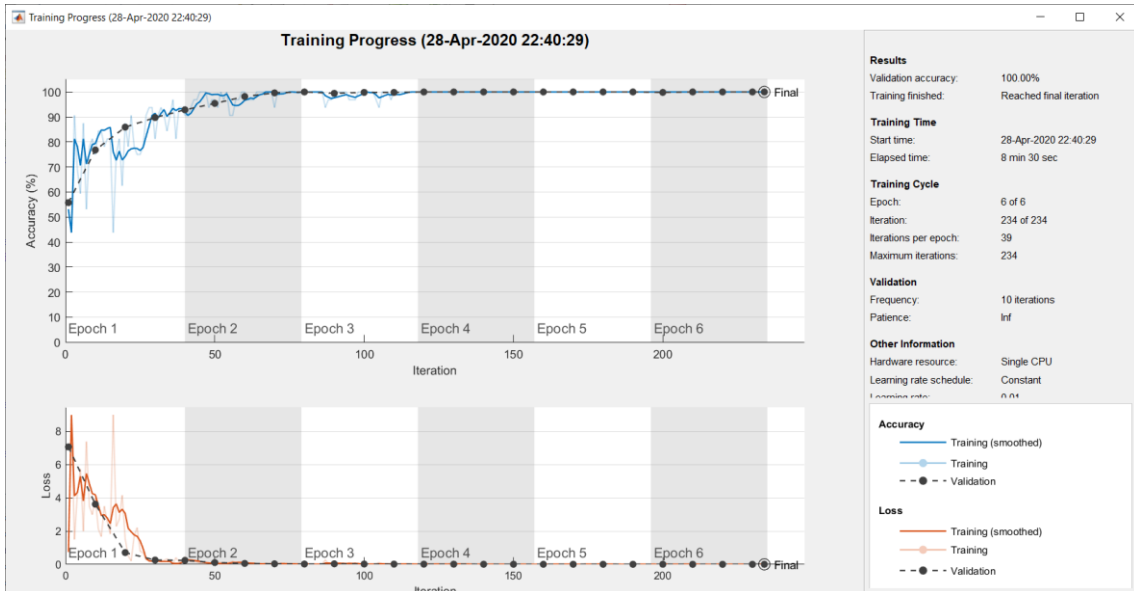


Imagen 120. Cámara Hospital3-Isleta.

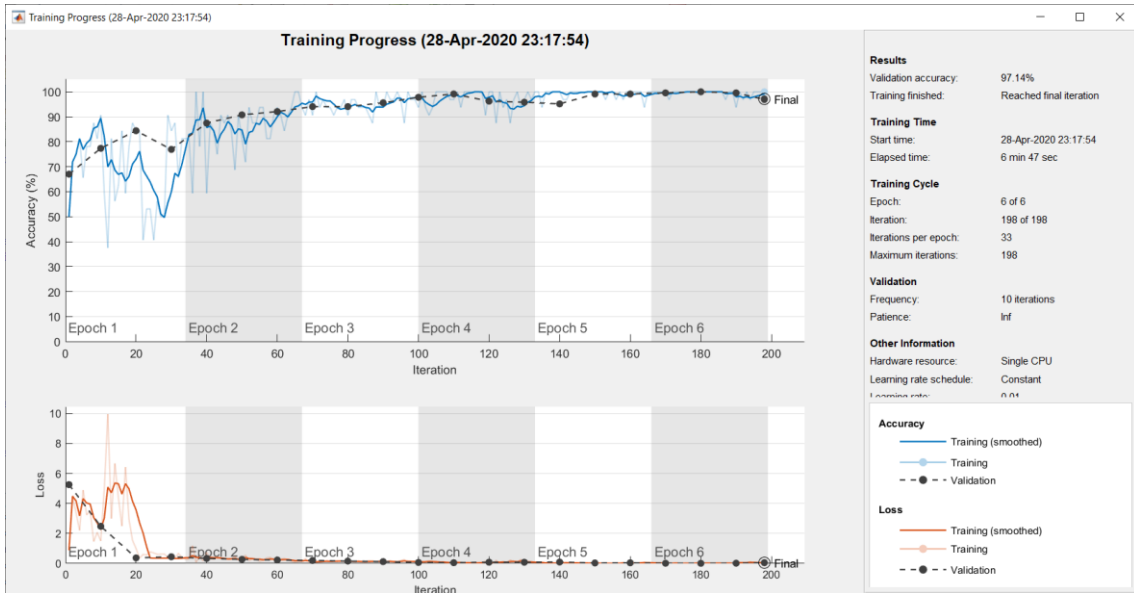


Imagen 121. Cámara Hospital4-Escaleritas.

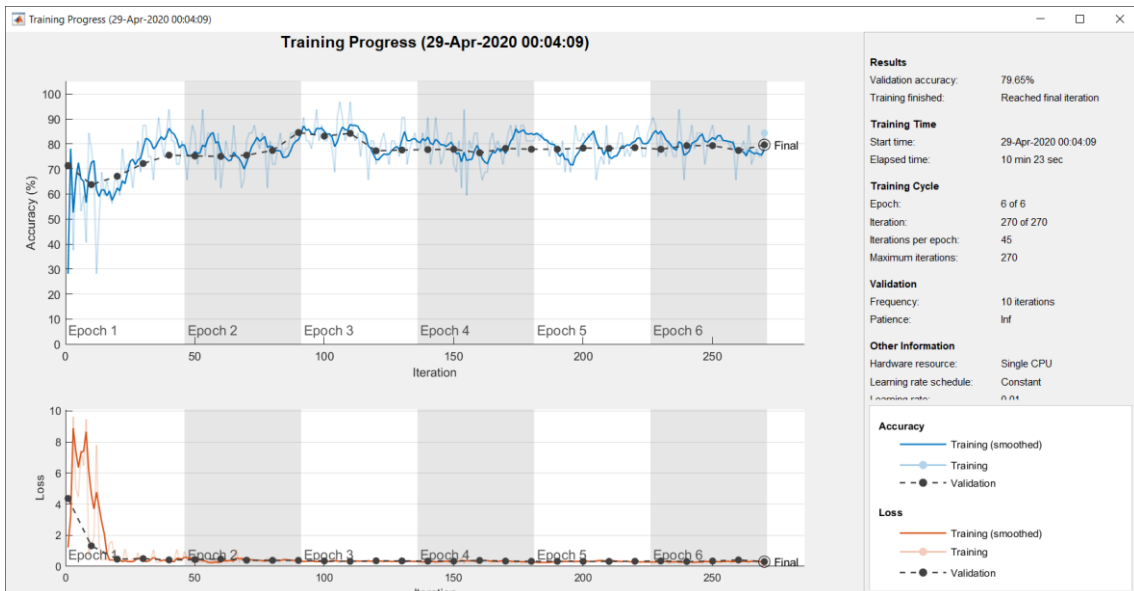


Imagen 122. Cámara Tafira1-Informática.

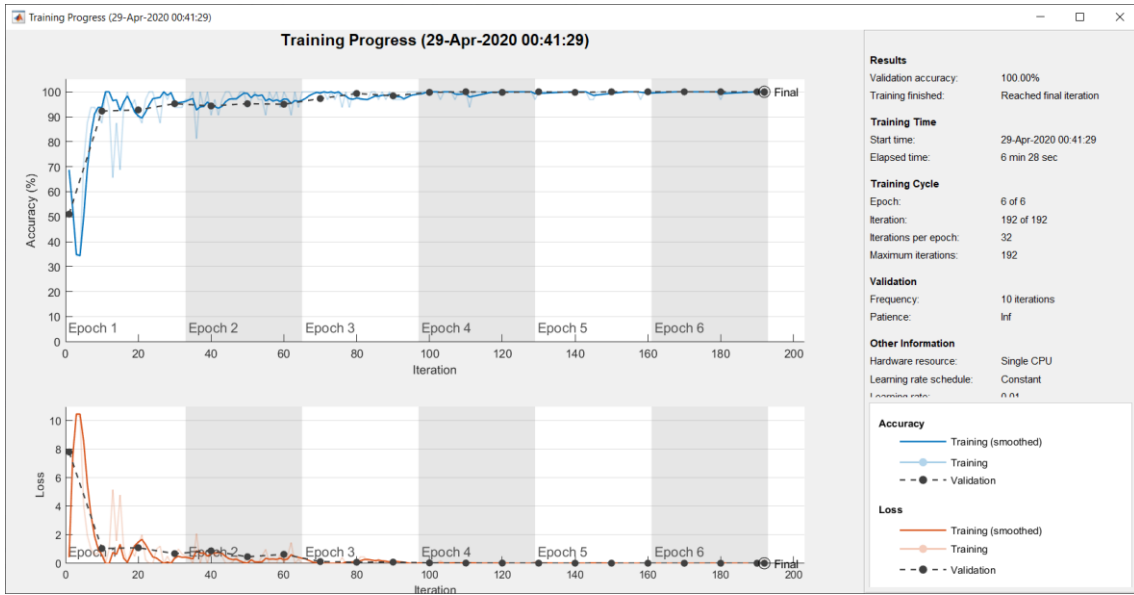


Imagen 123. Cámara Tafira2-MontañaArucas.

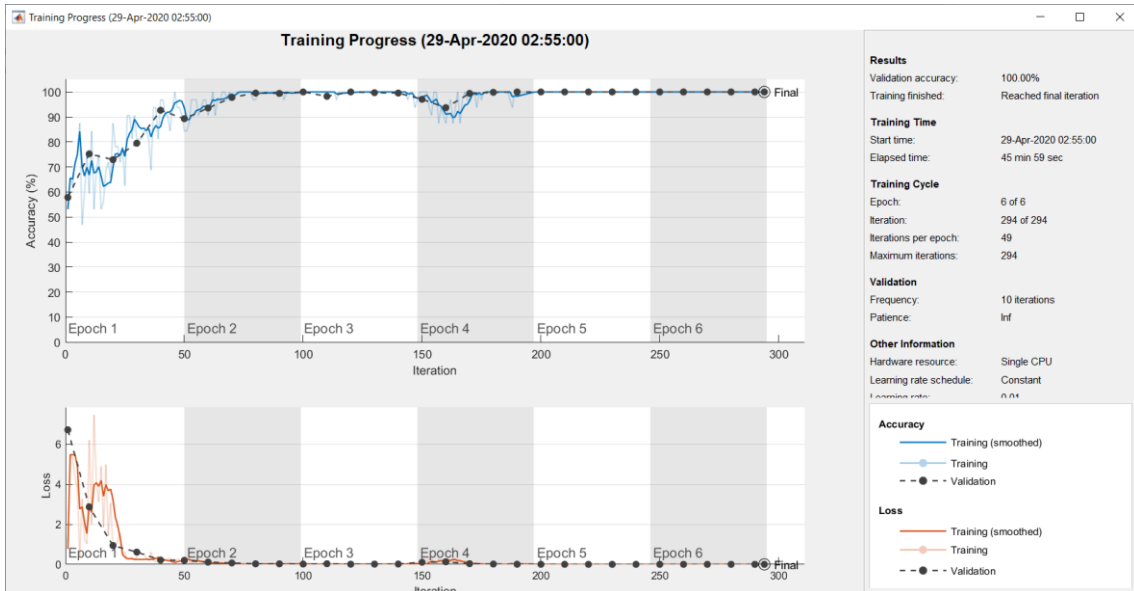


Imagen 124. Cámara Tafira3-Foscam.

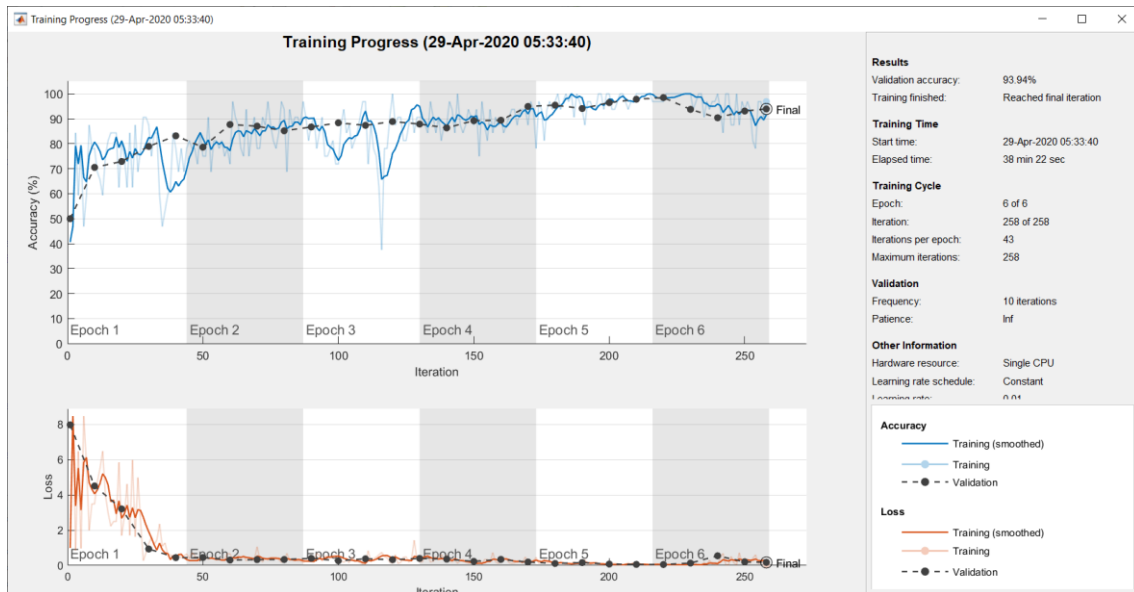


Imagen 125. Cámara Tafira4-Foscam.

6.4.6 Experimentos sobre cuantificación

Una vez realizada la detección se procedió a realizar la cuantificación.

Para realizar la cuantificación modificaremos la clasificación de la red neuronal convolucional de la detección anterior. Por ello, para medir los parámetros de calidad para la cuantificación es necesario ampliar la matriz de confusión anterior a un tamaño de 3x3 como se ha explicado en capítulos anteriores.

En este momento disponemos de 3 clases en el aprendizaje supervisado de la red neuronal:

- **NL_NN:** No hay datos meteorológicos reales de lluvia y el nivel de precipitación es nulo.
- **SL_NB:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es bajo.
- **SL_NA:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es alto.

Realizamos experimentos con 1, 2 y 3 capas intermedias u ocultas en la red. Y observamos que, a partir de la tercera capa, los resultados no mejoraban significativamente. Por esta razón sólo añadimos 3 capas intermedias.

Un aprendizaje supervisado costa de dos etapas, una de test y otra de entrenamiento. A la red neuronal se le aporta una base de datos donde se dispone de las dos clasificaciones de las imágenes antes descritas y, con ellas, comienza el proceso de aprendizaje máquina.

El diagrama de bloque es el ilustrado en la imagen 126:

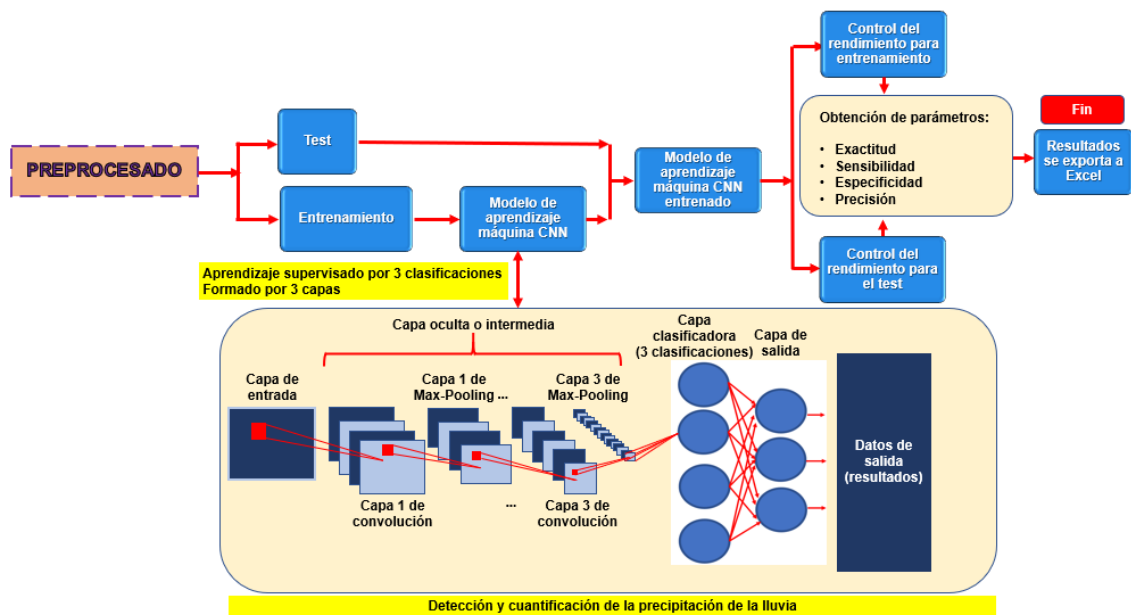


Imagen 126. Diagrama de bloques incluyendo la etapa de la red neuronal convolucional.

Los resultados obtenidos son los siguientes:

- Resultados usando el modo de color Cielab, toda la etapa de procesamiento descrita con anterioridad y mediante un aprendizaje supervisado por 3 clasificaciones y 3 capas intermedias.

A continuación, se muestra las gráficas de la primera iteración.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 214 | 1 | 0 |
| 3 | SL_NB | 0 | 216 | 0 |
| 4 | SL_NA | 0 | 9 | 0 |

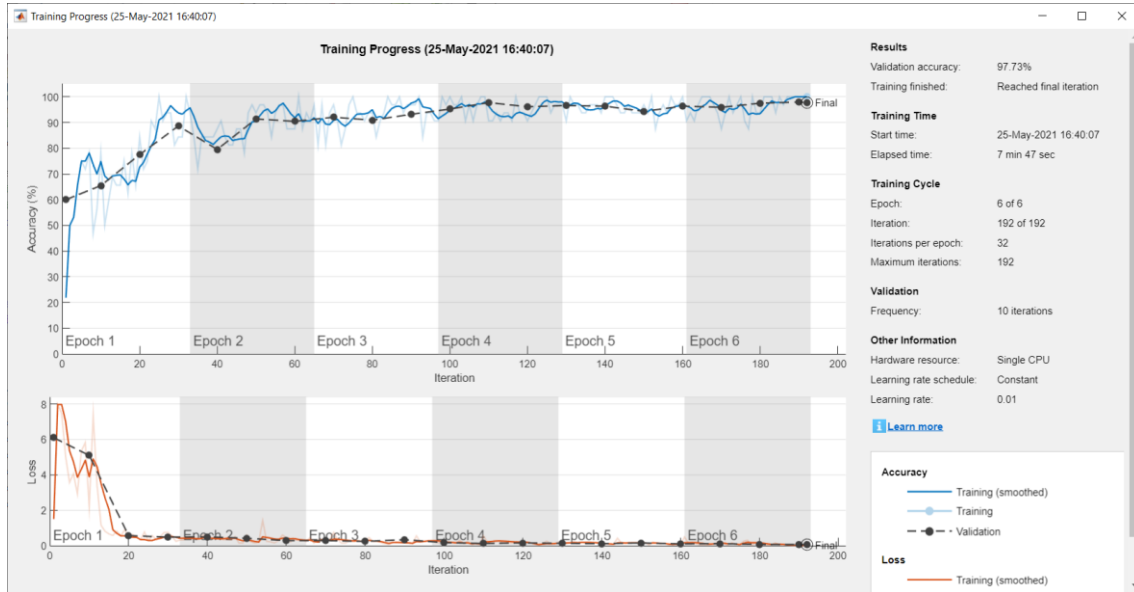


Imagen 127. Cámara Hospital1-Barranoseco.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 144 | 0 | 0 |
| 3 | SL_NB | 25 | 154 | 0 |
| 4 | SL_NA | 0 | 16 | 20 |

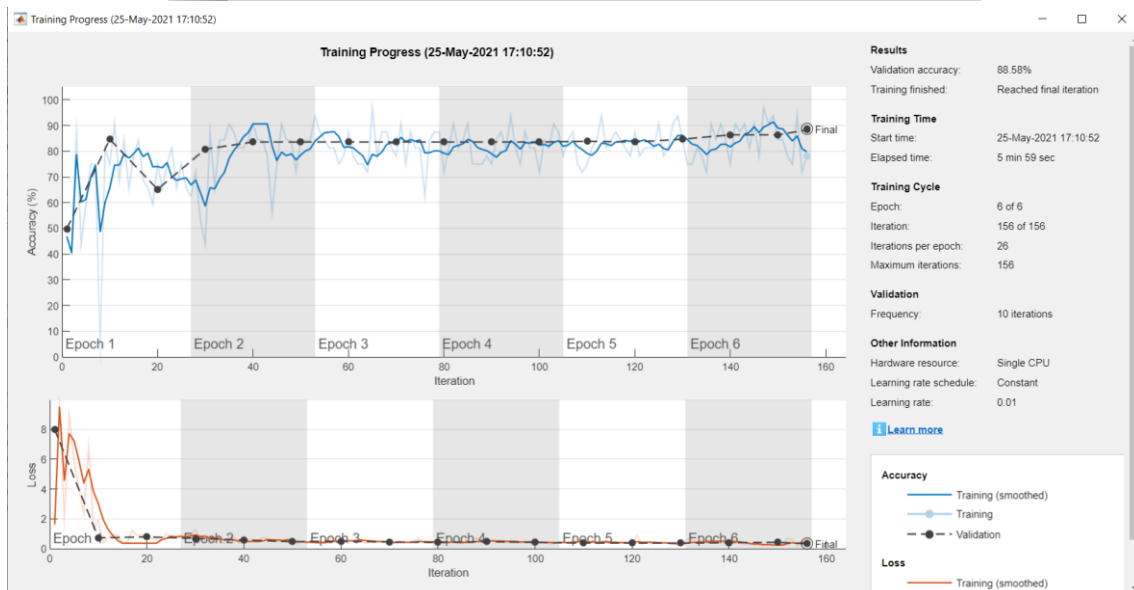


Imagen 128. Cámara Hospital2-Catedral.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 305 | 0 | 0 |
| 3 | SL_NB | 0 | 134 | 0 |
| 4 | SL_NA | 0 | 0 | 36 |

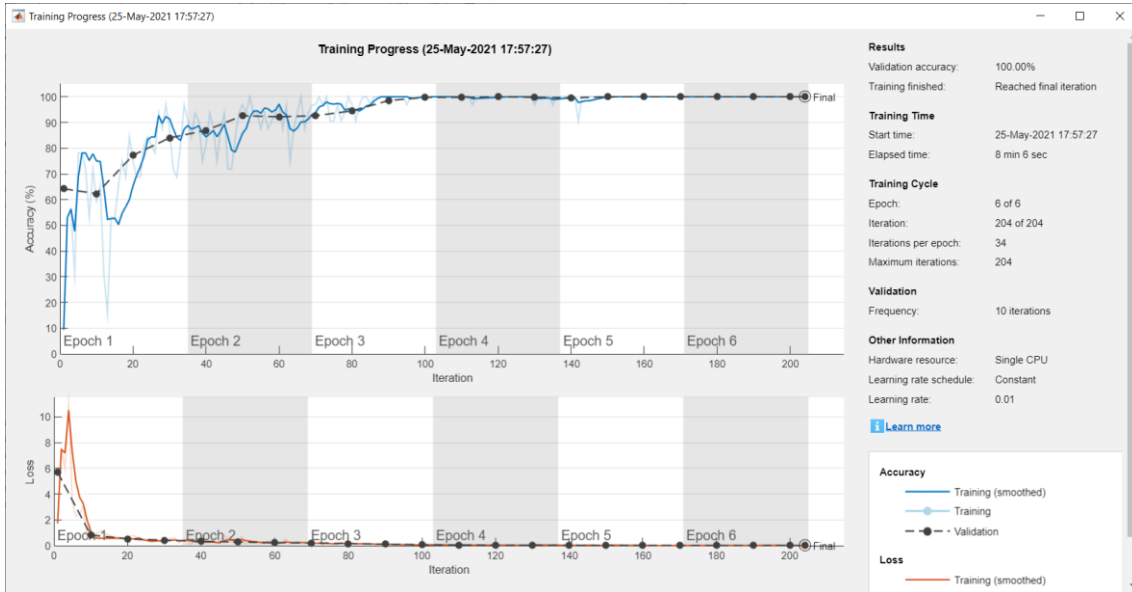


Imagen 129. Cámara Hospital3-Isleta.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 232 | 0 | 0 |
| 3 | SL_NB | 0 | 151 | 0 |
| 4 | SL_NA | 0 | 0 | 9 |

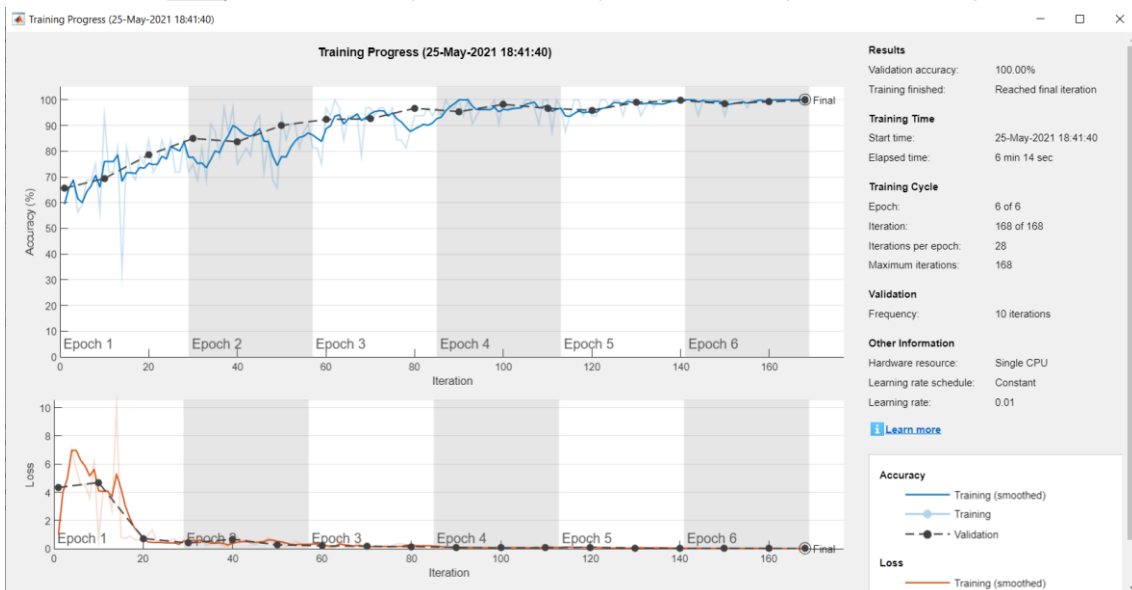


Imagen 130. Cámara Hospital4-Escaleritas.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 557 | 0 | 0 |
| 3 | SL_NB | 0 | 952 | 2 |
| 4 | SL_NA | 0 | 6 | 192 |

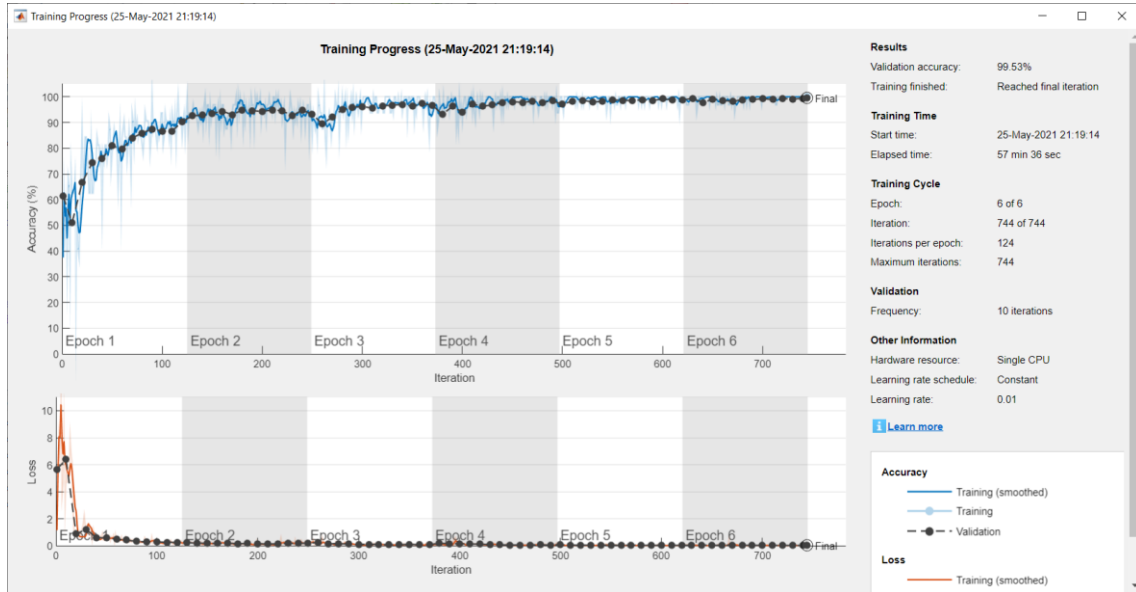


Imagen 131. Cámara Tafira1-Infornática.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 216 | 0 | 0 |
| 3 | SL_NB | 0 | 458 | 0 |
| 4 | SL_NA | 0 | 0 | 1080 |

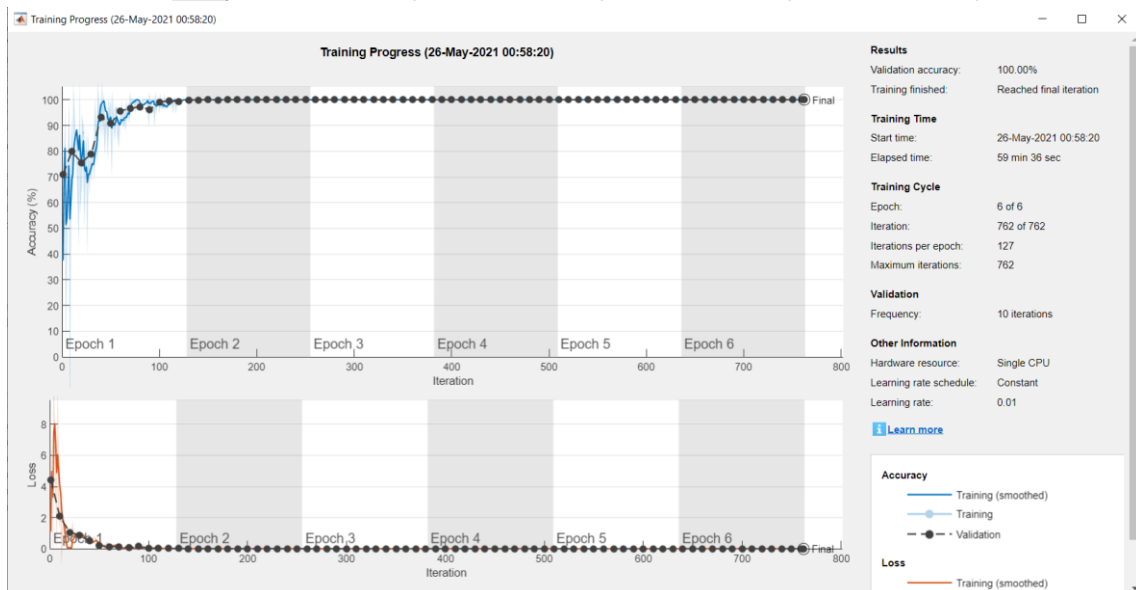


Imagen 132. Cámara Tafira2-MontañaArucas.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 288 | 0 | 0 |
| 3 | SL_NB | 20 | 830 | 13 |
| 4 | SL_NA | 0 | 0 | 792 |

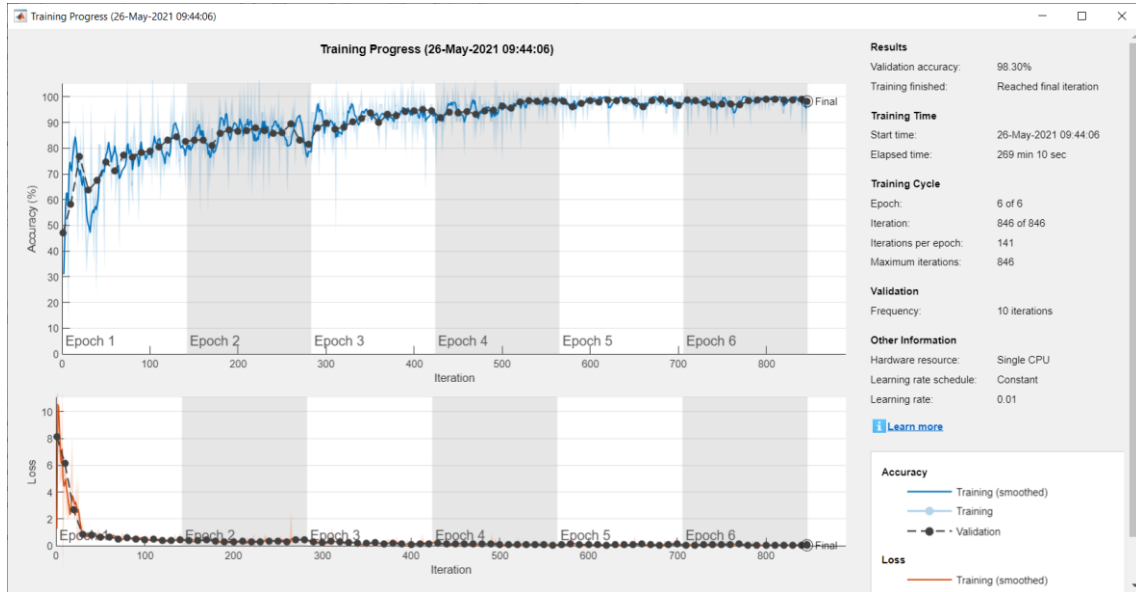


Imagen 133. Cámara Tafira3-Foscam.

6.4.7 Resultados de los intervalos de confianza

A continuación, mostraremos los resultados obtenidos por todo el proceso descrito en el TFM.

- Intervalo de confianza de la matriz binaria (Tabla 15).

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---------------------|------------------|---------------------|------------------------|---------------------|------------------------|-------------------------|----------------------|-------------------------|---------------------|------------------|---------------------|
| 1 | LI Exactitud | Exactitud | LS Exactitud | LI Sensibilidad | Sensibilidad | LS Sensibilidad | LI Especificidad | Especificidad | LS Especificidad | LI Precisión | Precisión | LS Precisión |
| 2 | 0,99 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 |
| 3 | 0,90 | 0,93 | 0,95 | 0,84 | 0,88 | 0,92 | 0,98 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 |
| 4 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 |
| 5 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 |
| 6 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 7 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 8 | 0,99 | 0,99 | 0,99 | 0,98 | 0,99 | 0,99 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |

Tabla 15. Intervalos de confianza de la matriz binaria.

Donde:

- **LS Exactitud:** límite superior del intervalo de confianza para la exactitud.
- **LI Exactitud:** límite inferior del intervalo de confianza para la exactitud.
- **Exactitud:** valor de exactitud obtenido.
- **LS Sensibilidad:** límite superior del intervalo de confianza para la sensibilidad.
- **LI Sensibilidad:** límite inferior del intervalo de confianza para la sensibilidad.
- **Sensibilidad:** valor de sensibilidad obtenido.
- **LS Precisión:** límite superior del intervalo de confianza para la precisión.
- **LI Precisión:** límite inferior del intervalo de confianza para la precisión.
- **Precisión:** valor de precisión obtenido.
- **LS Especificidad:** límite superior del intervalo de confianza para la especificidad.
- **LI Especificidad:** límite inferior del intervalo de confianza para la especificidad.
- **Especificidad:** valor de especificidad obtenido.

- Intervalos de confianza de la matriz de confusión (Tabla 16).

| | A | B | C | D | E | F | G | H | I |
|---|-----------------|--------------|-----------------|-----------------|--------------|-----------------|-----------------|--------------|-----------------|
| 1 | LI Exactitud NN | Exactitud NN | LS Exactitud NN | LI Exactitud NB | Exactitud NB | LS Exactitud NB | LI Exactitud NA | Exactitud NA | LS Exactitud NA |
| 2 | 0,98 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 0,00 | 0,00 | 0,00 |
| 3 | 0,98 | 1,00 | 1,00 | 0,81 | 0,86 | 0,90 | 0,42 | 0,56 | 0,68 |
| 4 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 | 0,93 | 1,00 | 1,00 |
| 5 | 0,99 | 1,00 | 1,00 | 0,98 | 1,00 | 1,00 | 0,77 | 1,00 | 1,00 |
| 6 | 1,00 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 0,94 | 0,97 | 0,98 |
| 7 | 0,99 | 1,00 | 1,00 | 0,99 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 8 | 0,99 | 1,00 | 1,00 | 0,95 | 0,96 | 0,97 | 1,00 | 1,00 | 1,00 |

Tabla 16. Intervalos de confianza de la matriz de confusión.

Donde:

- **LS Exactitud NN:** límite superior del intervalo de confianza para la exactitud del nivel nulo de lluvia.
- **LI Exactitud NN:** límite inferior del intervalo de confianza para la exactitud del nivel nulo de lluvia.
- **Exactitud NN:** valor de exactitud obtenido del nivel nulo de lluvia.
- **LS Exactitud NB:** límite superior del intervalo de confianza para la exactitud del nivel bajo de lluvia.
- **LI Exactitud NB:** límite inferior del intervalo de confianza para la exactitud del nivel bajo de lluvia.
- **Exactitud NB:** valor de exactitud obtenido del nivel bajo de lluvia.
- **LS Exactitud NA:** límite superior del intervalo de confianza para la exactitud del nivel alto de lluvia.
- **LI Exactitud NA:** límite inferior del intervalo de confianza para la exactitud del nivel alto de lluvia.
- **Exactitud NA:** valor de exactitud obtenido del nivel alto de lluvia.

Capítulo 7:

En este capítulo exponemos dos puntos:

1. Las conclusiones obtenidas en este TFM. Mostramos los objetivos planteados inicialmente y los resultados obtenidos de los diferentes experimentos realizados.
2. Propuesta de líneas de investigación futuras relacionadas con la implementación y desarrollos de sistemas de aprendizaje máquina para la detección y cuantificación de la lluvia sobre imágenes de videovigilancia.

7.1 Conclusiones

En el desarrollo de este TFM se han logrado alcanzar una serie de objetivos marcados al inicio del mismo, esos son:

1. Implementación de una metodología sobre la base de algoritmos de Machine Learning para la detección y cuantificación de la precipitación de la lluvia basado en imágenes de cámaras distribuidas por diferentes zonas de observación.
2. Se han estudiado las de las características necesarias de los dispositivos de captación de imágenes. Hemos verificado que la calidad de las imágenes del sistema VIMETRI es adecuada para nuestros propósitos.

3. Una vez determinadas las características de las cámaras e imágenes planteamos la adecuación del almacenamiento para nuestros propósitos, considerando los medios necesarios.
4. Estudio de los procesos previos a la aplicación de las imágenes. Para alcanzar una adecuada calidad de imagen para su posterior procesamiento se ha realizado un pre-procesado de la señal. Este pre-procesado incluye balance de blancos, filtro de Sobel, selección de regiones de interés (ROIs), etc.
5. Estudiamos algoritmos de reconocimiento de imágenes y detección de eventos. Particularmente, hemos implementado una metodología en la que se usan Redes Neuronales Convolucionales.

Asimismo, de los resultados obtenidos y presentados en el capítulo anterior, se extraen las siguientes conclusiones:

1. Mediante el uso de redes neuronales convolucionales se han detectado de forma satisfactoria las precipitaciones de lluvia mediante las imágenes recibidas desde cámaras de vigilancia.
2. Se ha podido cuantificar su nivel de precipitación mediante 3 clasificaciones: nula, bajo, alto.
3. Para valorar el grado de fiabilidad de los datos obtenidos en la detección y la calidad de cuantificación, hemos calculado el intervalo o margen de confianza al 95%. Y, mediante la observación del límite superior e inferior, al calcular la diferencia de ambos valores podemos verificar que la credibilidad de los datos obtenidos es muy satisfactoria. Ya que la diferencia entre los límites es muy pequeña y, por lo tanto, mayor credibilidad poseen nuestros resultados.

En definitiva, dejamos sentadas las bases metodológicas y experimentales de un prototipo de detección y cuantificación de lluvia mediante imágenes de cámaras de video.

7.2 Líneas futuras

Las líneas de investigación que se proponen para trabajar en un futuro para la mejora de este TFM son:

- Estudiar los movimientos de las cámaras debido a la acción del viento sobre las mismas y el mástil. Realizar un algoritmo que corrija dicha problemática.
- Adaptar el código sin discernir entre diferentes cámaras.
- Aplicar diferentes técnicas para calcular el gradiente en las imágenes.
- Aplicar otras arquitecturas de redes neuronales, tales como LSTM (Long Short Term Memory). Así mismo, en los trabajos específicos sobre imágenes de videovigilancia está el TFG de Darío Díaz [Díaz 2021], que es continuación de este TFM. En este TFG se dan resultados de detección con arquitecturas básicas de LSTM.
- Estudiar el movimiento de los elementos que aparecen en una imagen, tales como el movimiento de las palmeras.
- Identificación de calima.

Anexo I

Implementación del código en Matlab

En este anexo se procederá a comentar los algoritmos empleados en el entorno Matlab en este TFM.

Fragmento 1 del algoritmo:

```
5      % PARÁMETROS DEL PROGRAMA, COMUNES A TODOS LOS EXPERIMENTOS
6      %_____training_____
7 -    Min_Frames = 30;
8      % Filtrado de Sobel
9 -    Filt_Sobel = 1;
10 -   Size_Sobel = 3;
11 -   [Mx,My]    = GSobel_Filters(Size_Sobel);
12 -   Borde_Sobel = fix(Size_Sobel/2);
```

Imagen 134. Fragmento 1 del algoritmo implementado en matlab.

Línea 7:

Se detalla uno de los parámetros de calidad de los datos recogidos en este TFM. Este TFM se ha particularizado a recoger datos con un número de imágenes o fotogramas limitados:

- **Número de fotogramas/segundo:** 30 fotogramas/segundo.

Línea 9:

Se ha incluido una variable que activa o desactiva el filtro de SOBEL:

- **Filt_Sobel=1:** si se realiza el filtrado SOBEL.
- **Filt_Sobel=0:** no se realiza el filtrado SOBEL.

Línea 10:

Se introduce el tamaño de los filtros de SOBEL.

Línea 11:

Realización de la llamada de la función que realiza el filtrado de SOBEL. Sus parámetros de entrada y salida son los siguientes:

- **Parámetros de entrada:** tamaño de los filtros de SOBEL.
- **Parámetros de salida:** máscara horizontal y vertical.

La función llamada es la siguiente:

```
15 function [Mx,My]= GSobel_Filters(Wsize)
16 for i=1:Wsize
17     Sx(i)=factorial((Wsize-1))/((factorial((Wsize-1)-(i-1)))*(factorial(i-1)));
18     Dx(i)=Pasc(i-1,Wsize-2)-Pasc(i-2,Wsize-2);
19 end
20 Sy=Sx';
21 Dy=Dx';
22 Mx=Sy(:)*Dx;
23 My=Mx';
24
25 function P=Pasc(k,n)
26 if (k>=0)&&(k<=n)
27     P=factorial(n)/(factorial(n-k)*factorial(k));
28 else
29     P=0;
30 end
```

Imagen 135. Función del filtrado de Sobel.

Esta función se ha explicado con anterioridad en el capítulo 4, en concreto, apartado 4.4.

Línea 12:

Posteriormente en el algoritmo realizaremos la convolución de las imágenes, en este proceso la cantidad de píxeles que se necesitan es inferior a la calculada en los pasos anteriores. Por ello, se procede a reducir el tamaño de los filtros de SOBEL.

Fragmento 2 del algoritmo:

```

14 % Excel matriz binaria
15 numeroExperimento= 'Cuantificacion_Matriz_Binaria';
16 filename= strcat('Resultados_', numeroExperimento, '.xlsx');
17 A = ['Accuracy (Exactitud)', 'Sensibilidad (Sensibilidad)', 'Specificity (Especificidad)', 'Precision (Precisión)'];
18 B = {'TN', 'FP', 'FN', 'TP'};
19 Rango1 = 'A1:D1';
20 Rango2 = 'A12:D12';
21 % Fin matriz binaria
22
23 % Excel matriz de confusión 3x3 cuantificación
24 numeroExperimento2= 'Cuantificacion_Matriz_Confesion';
25 filename2= strcat('Resultados_', numeroExperimento2, '.xlsx');
26 C = {'NL_NN', 'SL_NB', 'SL_NA'};
27 D = {'NL_NN', 'SL_NB', 'SL_NA'};
28 hoja2=1;
29 Rango3 = 'B1:D1';
30 Rango4 = 'A2:A4';
31 xlswrite(filename2,C, hoja2, Rango3);
32 xlswrite(filename2,D, hoja2, Rango4);
33 % FIN Excel matriz de confusión 3x3 cuantificación
34
35 % Excel Nivel de Confianza matriz binaria
36 filename3= 'Resultados_Nivel_Confianza_Matriz_Binaria.xlsx';
37 E = {'LI Exactitud', 'Exactitud', 'LS Exactitud', 'LI Sensibilidad', 'Sensibilidad', 'LS Sensibilidad', 'LI Especificidad', 'Especificidad', 'LS Especificidad', 'LI Precisión', 'Precisión', 'LS Precisión'};
38 RangoNC = 'A1:I1';
39 % Fin Excel Nivel de Confianza matriz binaria
40
41 % Excel Nivel de Confianza matriz de confusión
42 filenameCMC= 'Resultados_Nivel_Confianza_Matriz_De_Confesion.xlsx';
43 tituloCMC = {'LI Exactitud NN', 'Exactitud NN', 'LS Exactitud NN', 'LI Exactitud NB', 'Exactitud NB', 'LS Exactitud NB', 'LI Exactitud NA', 'Exactitud NA', 'LS Exactitud NA'};
44 RangoCMC = 'A1:I1';
45 % Fin Excel Nivel de Confianza matriz de confusión

```

Imagen 136. Fragmento 2 del algoritmo implementado en matlab.

Líneas 15-20 y 35-39:

En estas líneas generamos un fichero Excel para guardar los datos de los parámetros de detección y otro fichero Excel para los márgenes de confianza calculados. Están explicados en el capítulo 6.

Líneas 23-33 y 41-45:

En estas líneas generamos un fichero Excel para guardar los datos de los parámetros de la matriz de confusión para la detección y, otro fichero para los márgenes de confianza calculados. Están explicados en el capítulo 6.

Una vez asignado un nombre al fichero Excel, incluimos los nombres de los parámetros de cuantificación para detallarlos en cada hoja del fichero.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 288 | 0 | 0 |
| 3 | SL_NB | 8 | 855 | 0 |
| 4 | SL_NA | 0 | 9 | 783 |

Imagen 137. Excel matriz de confusión.

Están explicados con mayor detalle en el capítulo 6.

Fragmento 4 del algoritmo:

```
49 - for r=1:30
50 -
51 -     inputFolderVideo = [pwd '\ ' Videovigilancia_Meteorologica' ];
52 -
53 -     Experimentos_Camaras = readtable('Camaras_NN.csv','Delimiter',';');
54 -
55 -
56 -     [Ic,Jc]      = size(Experimentos_Camaras);
57 -
58 -     DD          = dir(inputFolderVideo);
59 -     [ID,JD]     = size(DD);
60 -
61 -     Scores      = zeros(Ic,2,2);
62 -     Scores_Copia = zeros(1,4);
63 -     ScoresC     = zeros(Ic,2,2);
64 -     ScoresC_Copia = zeros(1,4);
65 -
66 -     % EXCEL MATRIZ BINARIA
67 -     hojal = r; %recuerda que cada hoja del excel guarda los resultados
68 -     xlswrite(filename,A,hojal,Rango1);
69 -     xlswrite(filename,B,hojal,Rango2);
70 -     filal=2; %luego dentro del bucle irá cambiando
71 -     fila2=13; %luego dentro del bucle irá cambiando
72 -     % FIN EXCEL MATRIZ BINARIA
73 -
74 -     % Excel Nivel de Confianza matriz binaria
75 -     hoja3=r;
76 -     xlswrite(filename3,E,hoja3,RangoNC);
77 -     fila3=2; %luego dentro del bucle irá cambiando
78 -     % FIN Excel Nivel de Confianza matriz binaria
79 -
80 -     % Excel Nivel de Confianza matriz de confusión
81 -     hojaNCMC=r;
82 -     xlswrite(filenameNCMC,tituloNCMC,hojaNCMC,RangoNCMC);
83 -     filaNCMC=2; %luego dentro del bucle irá cambiando
84 -     % FIN Excel Nivel de Confianza matriz de confusión
```

Imagen 138. Fragmento 4 del algoritmo implementado en matlab.

Línea 49:

Bucle utilizado para repetir los experimentos múltiples veces, de esta forma se puede estudiar la fiabilidad y calidad de los datos obtenidos.

Línea 51:

Acceso al directorio donde se encuentra ubicada la base de datos denominada “Vigilancia_Meteorologica” que contiene los vídeos recogidos de las cámaras de videovigilancia.

```
inputFolderVideo = [pwd '\ ' Videovigilancia_Meteorologica' ];
```

```
inputFolderVideo: 1x87 char =
D:\Naira\MASTER\ASIGNATURAS 23-04-2020\NAIRA\TFM\Naira SW\Videovigilancia_Meteorologica
```

Imagen 139. Acceso al directorio de los vídeos de las cámaras.

Línea 53:

Leemos el contenido de 'experimentos_Camaras.csv' y creamos una tabla con los datos leídos.

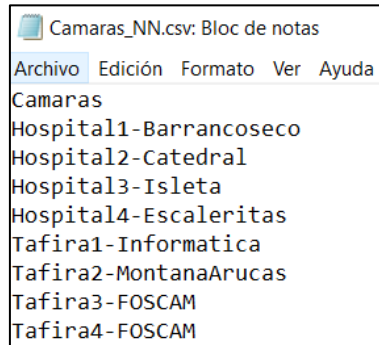


Imagen 140. Fichero txt con las cámaras usadas en el TFM.

```
Experimentos_Camaras = readtable('Camaras_NN.csv','Delimiter',';');
```

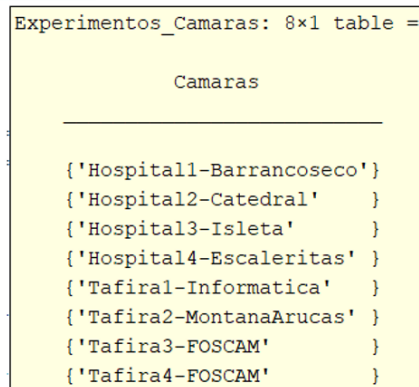


Imagen 141. Lectura de fichero txt anterior.

Los datos extraídos son 8, que representan los nombres de las cámaras de videovigilancia.

Línea 59:

Función que calcula el tamaño de la tabla de la línea 53. Los parámetros de salida son:

- **lc:** número de filas.
- **Jc:** números de columnas.

```

Command Window

K>> Ic

Ic =

     8

K>> Jc

Jc =

     1

```

Imagen 142. Valor de las filas y las columnas.

Línea 58:

Función de Matlab que muestra la lista de contenidos del directorio introducido como parámetro de entrada.

| Fields | name | folder | date | bytes | isdir | datenum |
|--------|--------------------------|---------------|----------------|-------|-------|------------|
| 1 | '' | 'D:\Naira\... | '02-oct.-20... | 0 | 1 | 7.3807e+05 |
| 2 | '' | 'D:\Naira\... | '05-nov.-20... | 0 | 1 | 7.3810e+05 |
| 3 | 'Hospital1-Barranoseco' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 4 | 'Hospital2-Catedral' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 5 | 'Hospital3-Isleta' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 6 | 'Hospital4-Escaleritas' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 7 | 'Tafira1-Informatica' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 8 | 'Tafira2-MontanaAruacas' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 9 | 'Tafira3-FOSCAM' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 10 | 'Tafira4-FOSCAM' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |

Imagen 143. Directorios los vídeos de cada cámara.

Líneas 61-64:

Inicialización de los vectores donde posteriormente guardaremos los resultados de los parámetros de detección: Exactitud, precisión, sensibilidad y especificidad.

Líneas 66-84:

Escribimos los resultados obtenidos en los ficheros descritos anteriormente.

Fragmento 4 del algoritmo:

```
87 -   for cc=1:Ic
88 -       Camara_t = Experimentos_Camaras(cc,1);
89 -       Camara   = Camara_t.Camaras{:};
90
91 -       inputFolderVideo = [pwd '\ ' 'Videovigilancia_Meteorologica' '\ ' Camara];
92 -       DD                = dir(inputFolderVideo);
93 -       [ID,JD]          = size(DD);
```

Imagen 144. Fragmento 4 del algoritmo implementado en matlab.

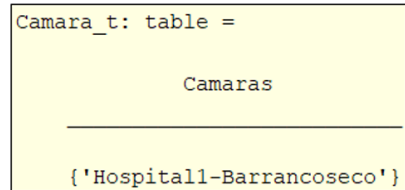
Línea 87:

Bucle que va seleccionando cámara a cámara de la extraída del fichero “Camaras_NN.csv”. La variable “cc” tomará valores desde 1 hasta 8.

Línea 88:

La variable “Camara_t” es una variable tipo tabla y en cada vuelta del bucle va adquiriendo un único nombre de una de las cámaras de videovigilancia.

```
Camara_t = Experimentos_Camaras(cc,1);
```



```
Camara_t: table =
           Camaras
-----
{'Hospital1-Barrancoseco'}
```

Imagen 145. Extracción del nombre de cada cámara.

Línea 89:

La variable “Camara” es una variable tipo cadena. En ella se guarda el nombre de la cámara en estudio.

Línea 91:

Directorio para acceder a la ubicación de una cámara específica en la base de datos “Vigilancia_Meteorologica”.

Línea 92:

Función de Matlab que muestra la lista de contenidos del directorio introducido como parámetro de entrada, en concreto el directorio de una cámara de videovigilancia.

| Fields | name | folder | date | bytes | isdir | datenum |
|--------|---------|---------------|----------------|-------|-------|------------|
| 1 | .. | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 2 | .. | 'D:\Naira\... | '02-oct.-20... | 0 | 1 | 7.3807e+05 |
| 3 | 'NL_NN' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 4 | 'SL_NA' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 5 | 'SL_NB' | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |

Imagen 146. Directorio de las subcarpetas de cada cámara.

Línea 93:

Función de Matlab que calcula el tamaño de la tabla obtenida en la línea 67. Sus parámetros de salida son:

- **ID:** número de filas, en este caso 5.
- **JD:** número de columnas, en este caso 1.

```
DD: 5x1 struct array with fields:
    name
    folder
    date
    bytes
    isdir
    datenum
```

Imagen 147. Variable DD representa el tamaño de la tabla.

Fragmento 5 del algoritmo:

```
94 -   for ii=1:(ID-2)
95 -       inputDir           = [inputFolderVideo '\ ' DD(ii+2).name];
96 -       DD2                = dir(inputDir);
97 -       [ID2,JD2]          = size(DD2);
```

Imagen 148. Fragmento 5 del algoritmo implementado en matlab.

Línea 94:

Bucle que va seleccionando de cada cámara una de las subcarpetas que contiene. Cada cámara consta de 3 subcarpetas:

- **NL_NN:** No hay nivel de lluvia.
- **SL_NA:** Nivel de lluvia alta.
- **SL_NB:** Nivel de lluvia baja.

Línea 95:

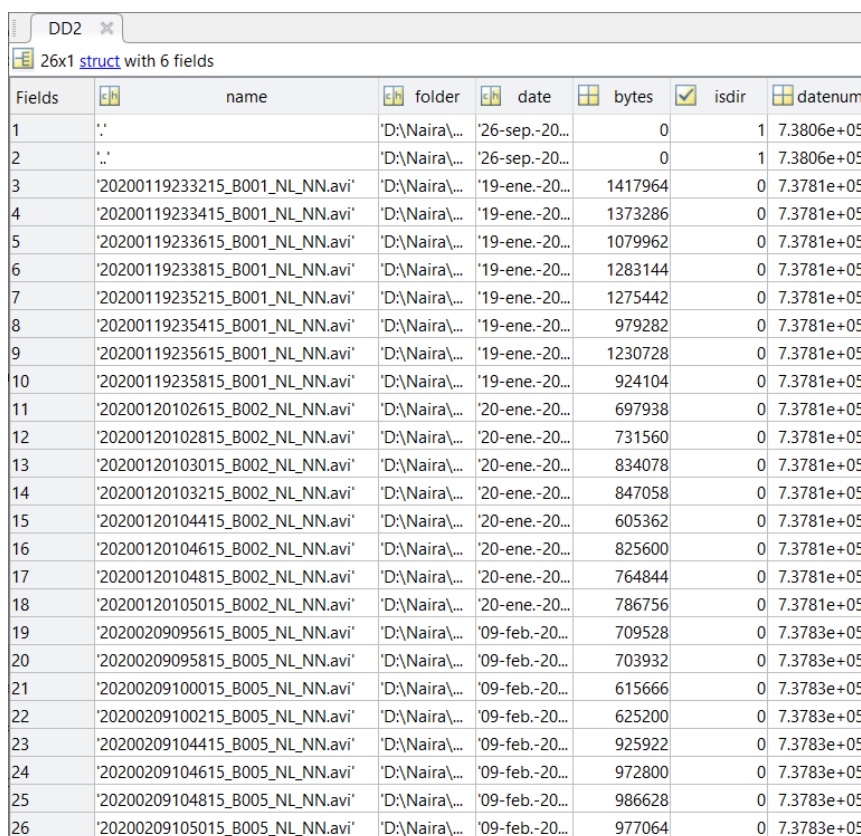
Directorio para acceder a la ubicación de una subcarpeta determinada de una cámara específica en la base de datos “Vigilancia_Meteorologica”.

```
Command Window
K>> inputDir
inputDir =
'D:\Naira\MASTER\ASIGNATURAS 23-04-2020\NAIRA\TFM\Naira SW\Videovigilancia_Meteorologica\Hospital1-Barranquenseco\NL_NN'
```

Imagen 149. Acceso a directorios de los vídeos.

Línea 96:

Función de Matlab que muestra la lista de contenidos del directorio introducido como parámetro de entrada, en concreto el directorio de una subcarpeta de una cámara de videovigilancia.



| Fields | name | folder | date | bytes | isdir | datenum |
|--------|---------------------------------|---------------|----------------|---------|-------|------------|
| 1 | .. | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 2 | .. | 'D:\Naira\... | '26-sep.-20... | 0 | 1 | 7.3806e+05 |
| 3 | '20200119233215_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 1417964 | 0 | 7.3781e+05 |
| 4 | '20200119233415_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 1373286 | 0 | 7.3781e+05 |
| 5 | '20200119233615_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 1079962 | 0 | 7.3781e+05 |
| 6 | '20200119233815_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 1283144 | 0 | 7.3781e+05 |
| 7 | '20200119235215_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 1275442 | 0 | 7.3781e+05 |
| 8 | '20200119235415_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 979282 | 0 | 7.3781e+05 |
| 9 | '20200119235615_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 1230728 | 0 | 7.3781e+05 |
| 10 | '20200119235815_B001_NL_NN.avi' | 'D:\Naira\... | '19-ene.-20... | 924104 | 0 | 7.3781e+05 |
| 11 | '20200120102615_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 697938 | 0 | 7.3781e+05 |
| 12 | '20200120102815_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 731560 | 0 | 7.3781e+05 |
| 13 | '20200120103015_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 834078 | 0 | 7.3781e+05 |
| 14 | '20200120103215_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 847058 | 0 | 7.3781e+05 |
| 15 | '20200120104415_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 605362 | 0 | 7.3781e+05 |
| 16 | '20200120104615_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 825600 | 0 | 7.3781e+05 |
| 17 | '20200120104815_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 764844 | 0 | 7.3781e+05 |
| 18 | '20200120105015_B002_NL_NN.avi' | 'D:\Naira\... | '20-ene.-20... | 786756 | 0 | 7.3781e+05 |
| 19 | '20200209095615_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 709528 | 0 | 7.3783e+05 |
| 20 | '20200209095815_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 703932 | 0 | 7.3783e+05 |
| 21 | '20200209100015_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 615666 | 0 | 7.3783e+05 |
| 22 | '20200209100215_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 625200 | 0 | 7.3783e+05 |
| 23 | '20200209104415_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 925922 | 0 | 7.3783e+05 |
| 24 | '20200209104615_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 972800 | 0 | 7.3783e+05 |
| 25 | '20200209104815_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 986628 | 0 | 7.3783e+05 |
| 26 | '20200209105015_B005_NL_NN.avi' | 'D:\Naira\... | '09-feb.-20... | 977064 | 0 | 7.3783e+05 |

Imagen 150. Directorio de cada video.

Línea 97:

Función que calcula el tamaño de la tabla de la línea 96. Los parámetros de salida son:

- **ID2:** número de filas, representa el número de vídeos que contiene esta subcarpeta.
- **JD2:** números de columnas.

```

Command Window
K>> ID2

ID2 =

    26

K>> JD2

JD2 =

     1

DD2: 26x1 struct array with fields:
    name
  folder
    date
   bytes
   isdir
  datenum
  
```

Imagen 151. Valor inicial de las variables ID2 y JD2.

Fragmento 6 del algoritmo:

```

98 - for jj=1:(ID2-2)
99 -     inputFileVideo      = DD2(jj+2).name;
100 -     inputFullFileVideo  = fullfile(inputDir, inputFileVideo);
101 -     obj                  = VideoReader(inputFullFileVideo);
102 -     video                = obj.read();
103 -     NumerodeFrames      = obj.NumFrames;
104 -     NumerodeFrames      = min(Min_Frames, NumerodeFrames);
  
```

Imagen 152. Fragmento 6 del algoritmo implementado en matlab.

Línea 98:

Bucle que va seleccionando cada vídeo de cada una de las subcarpetas de cada cámara.

Línea 99:

La variable “inputFileVideo” es una variable tipo cadena que contiene el nombre de un vídeo seleccionado por el bucle de la línea 98.

```

Command Window
K>> inputFileVideo

inputFileVideo =

    '20200119233215_B001_NL_NN.avi'
  
```

Imagen 153. Formación de una cadena con el nombre del vídeo.

Línea 100:

Directorio para acceder a la ubicación de un vídeo en una subcarpeta determinada de una cámara específica en la base de datos “Vigilancia_Meteorologica”.

```
Command Window
K>> inputFullFileVideo
inputFullFileVideo =
'D:\Naira\MASTER\ASIGNATURAS 23-04-2020\NAIRA\TFM\Naira SW\Videovigilancia_Meteorologica\Hospital1-Barrancoseco\NL_NN\20200119233215_B001_NL_NN.avi'
```

Imagen 154. Acceso a un vídeo.

Línea 101:

Función en Matlab que extrae todas las propiedades del vídeo en estudio y las guarda en la variable de salida denominada “obj”.

```
obj = VideoReader(inputFullFileVideo);
obj: 1x1 VideoReader =
VideoReader with properties:
General Properties:
    Name: '20200119233215_B001_NL_NN.avi'
    Path: 'D:\Naira\MASTER\ASIGNATURAS 23-04-2020\NAIRA\TFM\Naira SW\Videovigilancia_Meteorologica\Hospital1-Barrancoseco\NL_NN'
    Duration: 2.1000
    CurrentTime: 2.1000
    NumFrames: 42
Video Properties:
    Width: 1280
    Height: 720
    FrameRate: 20
    BitsPerPixel: 24
    VideoFormat: 'RGB24'
```

Imagen 155. Extracción de las propiedades de un vídeo.

Línea 102:

Lectura de los fotogramas del vídeo en estudio.

Línea 103:

Lectura del número de fotogramas/segundo del vídeo en estudio.

Línea 104:

Función en Matlab que calcula el número mínimo de fotogramas/segundo del vídeo.

Fragmento 7 del algoritmo:


```
106 -  for frame = 1 : NumerodeFrames
107
108 -     archivoImagen     = video(:,:,frame);
```

Imagen 156. Fragmento 7 del algoritmo implementado en matlab.

Línea 106:

Bucle que va seleccionando cada uno de los fotogramas de la imagen.

Línea 108:

Esta función de Matlab va seleccionando cada frame de la imagen y la guarda en la variable “archivoImangen”.

Fragmento 8 del algoritmo:

```
118 - switch Camara
119 -     case 'Tafiral-Informatica'
120 -         % CIELAB:
121 -         opcion='C';
122 -     case 'Tafira3-FOSCAM'
123 -         % CIELAB:
124 -         opcion='C';
125 -     case 'Hospital1-Barrancoseco'
126 -         % CIELAB:
127 -         opcion='C';
128 -     case 'Hospital2-Catedral'
129 -         % CIELAB:
130 -         opcion='C';
131 -     case 'Hospital3-Isleta'
132 -         % CIELAB:
133 -         opcion='C';
134 -     case 'Hospital4-Escaleritas'
135 -         % CIELAB:
136 -         opcion='C';
137 -     case 'Tafira2-MontanaArucas'
138 -         % CIELAB:
139 -         opcion='C';
140 -     case 'Tafira4-FOSCAM'
141 -         % CIELAB:
142 -         opcion='C';
143 -     otherwise
144 -         %si es otra cámara se coje por defecto CIELAB
145 -         opcion='C';
146 - end
```

Imagen 157. Fragmento 8 del algoritmo implementado en matlab.

Líneas 118-146:

Estructura condicional de Matlab, para seleccionar, según la cámara en estudio, un filtro de modo de color:

- **opcion= 'C'**: el modo de color seleccionado es Cielab.
- **opcion= 'L'**: el modo de color seleccionado es Luminancia.

Fragmento 9 del algoritmo:

```
148 - switch opcion
149 -     case {'l','L'}
150 -         % LUMINANCIA
151 -         %Pasamos la imagen tipo Pal a luminancia:
152 -         luminancia= 0.3*archivoImagen(:,:,1) + 0.59*archivoImagen(:,:,2) + 0.11*archivoImagen(:,:,3);
153 -
154 -         % Guardamos la imagen en luminancia:
155 -         Dir_Luminancia = ['Luminancia' '\\' Camara '\\' DD(ii+2).name];
156 -         DD_Luminancia = dir(Dir_Luminancia);
157 -         if ( isempty(DD_Luminancia) )
158 -             mkdir(Dir_Luminancia);
159 -         end
160 -
161 -         outputBaseFileName = sprintf('Frame_%4.4d_video_%d.png', frame,jj);
162 -         outputFullFileName = fullfile(Dir_Luminancia, outputBaseFileName);
163 -         imwrite(luminancia, outputFullFileName, 'png');
```

Imagen 158. Fragmento 9 del algoritmo implementado en matlab.

Línea 148:

Estructura condicional 'según el caso', según el caso seleccionado en las líneas 118-146 trabajaremos con el modo de color luminancia o Cielab.

Línea 152:

En esta línea procedemos a realizar el filtrado de modo de color, pasamos las imágenes de modo de color PAL a luminancia. Detallado en el capítulo 6.

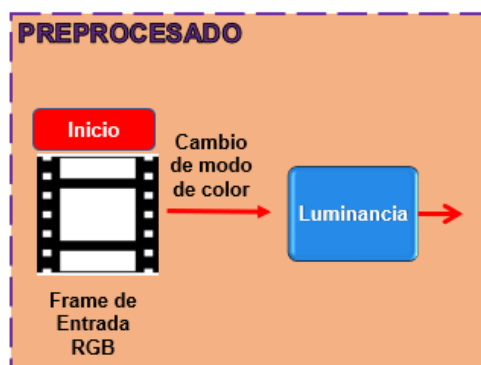


Imagen 159. Filtrado de modo de color.

Línea 155:

Creamos el directorio donde se van a ir almacenando cada una de las imágenes que se han filtrado con el modo de color de luminancia.

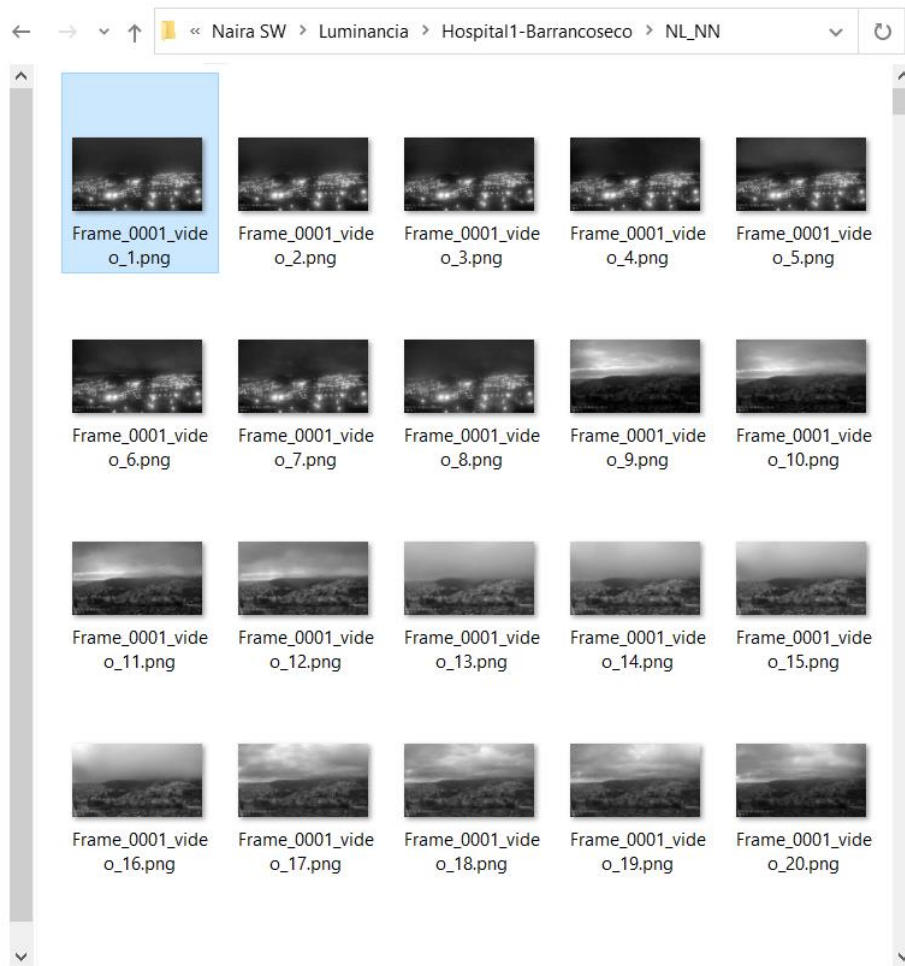


Imagen 160. Ejemplo de almacenamiento de los vídeos de las cámaras.

Línea 156:

Hacemos una copia del directorio donde se van a ir almacenando las imágenes en modo de color luminancia.

Líneas 157-159:

Estructura condicional que verifica si este directorio existe con anterioridad o no. En el caso de que no exista este directorio lo creamos por primera vez.

Línea 161:

Le asignamos un nombre a la imagen antes de guardarla en su directorio correspondiente. Se le asigna un número a la imagen y el número del vídeo al que pertenece. Y se guarda en formato 'png'.



Frame_0001_video_1.png

Imagen 161. Asignación de nombre de cada imagen tras pasar el primer preprocesado.

Línea 162:

Mediante la función 'fullfile' de Matlab creamos la ruta donde se va a ir almacenando las imágenes de modo de color luminancia.

Línea 163:

Se procede a guardar cada una de las imágenes en modo de color de luminancia en la ruta generada en la línea 162.

Fragmento 10 del algoritmo:

```
165 % Selecciono una sección de la imagen de luminancia:
166 - I = imread(outputFullFileName, 'png');
167 - [fil,col] = size(I);
168
169 % Para seleccionar el segmentado de imagen según la cámara
170 - switch Camara
171 -     case 'Tafiral-Informatica'
172 -         % Segmento n° 8:
173 -         I2 = imcrop(I, [(col-1)/3 ((fil-1)/3)*2 ((col-1)/3) (fil-1)/3]);
174 -     case 'Tafira3-FOSCAM'
175 -         % Segmento n° 8:
176 -         I2 = imcrop(I, [(col-1)/3 ((fil-1)/3)*2 ((col-1)/3) (fil-1)/3]);
177 -     case 'Hospital1-Barranoseco'
178 -         % Segmento n° 4:
179 -         I2 = imcrop(I, [0 (fil-1)/3 ((col-1)/3) (fil-1)/3]);
180 -     case 'Hospital2-Catedral'
181 -         % Segmento n° 9:
182 -         I2 = imcrop(I, [((col-1)/3)*2 ((fil-1)/3)*2 ((col-1)/3) (fil-1)/3]);
183 -     case 'Hospital3-Isleta'
184 -         % Segmento n° 9:
185 -         I2 = imcrop(I, [((col-1)/3)*2 ((fil-1)/3)*2 ((col-1)/3) (fil-1)/3]);
186 -     case 'Hospital4-Escaleritas'
187 -         % Segmento n° 8:
188 -         I2 = imcrop(I, [(col-1)/3 ((fil-1)/3)*2 ((col-1)/3) (fil-1)/3]);
189 -     case 'Tafira2-MontanaArucas'
190 -         % Segmento n° 5:
191 -         I2 = imcrop(I, [(col-1)/3 (fil-1)/3 ((col-1)/3) (fil-1)/3]);
192 -     case 'Tafira4-FOSCAM'
193 -         % Segmento n° 4:
194 -         I2 = imcrop(I, [0 (fil-1)/3 ((col-1)/3) (fil-1)/3]);
195 -     otherwise
196 -         %si es otra cámara se coje por defecto segmento n°5
197 -         I2 = imcrop(I, [(col-1)/3 (fil-1)/3 ((col-1)/3) (fil-1)/3]);
198 - end
```

Imagen 162. Fragmento 10 del algoritmo implementado en matlab.

Línea 166:

En esta línea se va leyendo cada una de las imágenes almacenadas en la carpeta de luminancia.

Línea 167:

Calculamos el tamaño de la imagen seleccionada anteriormente.

Líneas 170-198:

Estructura condicionante, dependiendo de la cámara se procede a seleccionar una región de interés.

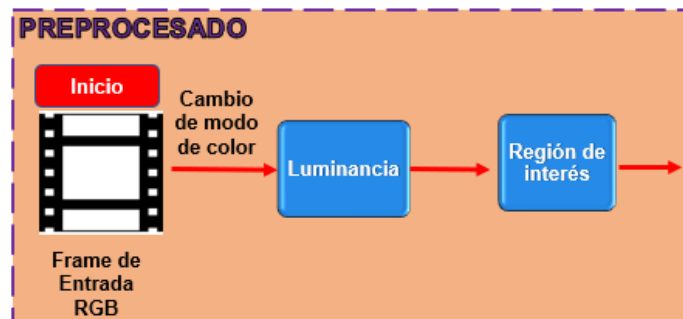


Imagen 163. Selección de las regiones de interés.

Como ya se ha comentado en el capítulo 4 de este TFM, para un mejor estudio de la imagen se le ha aplicado una sectorización para seleccionar una región de interés (Region of Interest, ROI). En cada imagen se pueden seleccionar 9 posibles regiones de interés como se muestra en la siguiente imagen.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Imagen 164. División de la imagen en diferentes regiones.

Después de un análisis exhaustivo de las imágenes de todas las cámaras hemos comprobado que podemos trabajar con 9 regiones de interés posibles, que nos permiten dos cosas:

- Focalizar zonas de mayor contraste de presencia de ruido.
- Identificar las zonas geográficas de mayor interés.

Con este filtrado podemos seleccionar de forma estratégica la sección de imagen donde realizaremos el estudio de la detección y cuantificación de la precipitación de la lluvia.

Se puede ver con más detalle en el capítulo 6.

Fragmento 11 del algoritmo:

```
201 % Guardo imagen de luminancia segmentada/recortada:
202 - Dir_Luminancia2 = ['Segmentación Luminancia' '\\' Camara '\\' DD(ii+2).name];
203 - DD_Luminancia2 = dir(Dir_Luminancia2);
204 - if ( isempty(DD_Luminancia2) )
205 -     mkdir(Dir_Luminancia2);
206 - end
207
208 - outputBaseFileName = sprintf('Frame_%4.4d_video_%d.png', frame,jj);
209 - outputFullFileName = fullfile(Dir_Luminancia2, outputBaseFileName);
210 - imwrite(I2, outputFullFileName, 'png');
```

Imagen 165. Fragmento 11 del algoritmo implementado en matlab.

Línea 202:

Creamos el directorio donde se van a ir almacenando cada una de las imágenes que se han filtrado con el modo de color de luminancia una vez se ha seleccionado la región de interés.

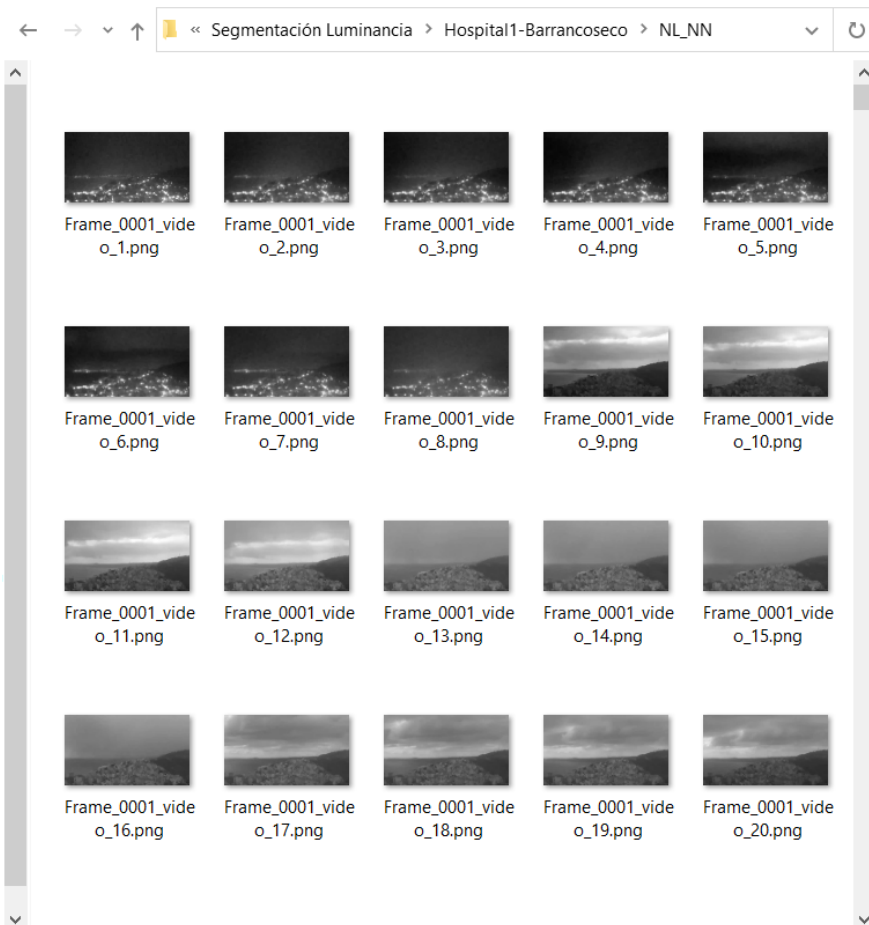


Imagen 166. Directorio donde se almacenan las imágenes de luminancia.

Línea 203:

Hacemos una copia del directorio donde se van a ir almacenando las imágenes en modo de color luminancia con la región de interés seleccionada.

Líneas 204-206:

Estructura condicional que verifica si este directorio existe con anterioridad o no. En el caso de que no exista este directorio lo creamos por primera vez.

Línea 208:

Le asignamos un nombre a la imagen antes de guardarla en su directorio correspondiente. Se le asigna un número a la imagen y el número del vídeo al que pertenece. Y se guarda en formato 'png'.



Frame_0001_video_1.png

Imagen 167. Nombre con el que se van almacenando las imágenes.

Línea 209:

Mediante la función 'fullfile' de Matlab creamos la ruta donde se va a ir almacenando las imágenes de modo de color luminancia con la región de interés seleccionada.

Línea 210:

Se procede a guardar cada una de las imágenes en modo de color de luminancia con la región de interés seleccionada en la ruta generada en la línea 184.

Fragmento 12 del algoritmo:

```
213 | % Filtrado de Sobel-----
214 | if (Filt_Sobel==1)
215 |     [Mx,My] = GSobel_Filters(3);
216 |     Cx      = conv2(I2,Mx);
217 |     Cy      = conv2(I2,My);
218 |     I2      = max(Cx(Borde_Sobel+1:end-Borde_Sobel,Borde_Sobel+1:end-Borde_Sobel),Cy(Borde_Sobel+1:end-Borde_Sobel,Borde_Sobel+1:end-Borde_Sobel));
219 | end
220 |
221 | % Guardo imagen del filtro Sobel:
222 | Dir_Luminancia3 = ['Segmentación Luminancia y SOBEL' '\ Camara '\ DD(ii+2).name];
223 | DD_Luminancia3 = dir(Dir_Luminancia3);
224 | if ( isempty(DD_Luminancia3) )
225 |     mkdir(Dir_Luminancia3);
226 | end
227 |
228 | outputBaseFileName = sprintf('Frame_%4.4d_video_%d.png', frame,jj);
229 | outputFullFileName = fullfile(Dir_Luminancia3, outputBaseFileName);
230 | imwrite(I2, outputFullFileName, 'png');
```

Imagen 168. Fragmento 12 del algoritmo implementado en matlab.

Líneas 213-219:

A las imágenes segmentadas se le aplica el filtrado de SOBEL.

Como ya se ha detallado en el capítulo 4, el gradiente de una función de dos variables (en este caso, la función de intensidad de la imagen) para cada punto es un vector bidimensional cuyos componentes están dados por las primeras derivadas de las direcciones verticales y horizontales. Recordemos que un borde es una discontinuidad, y para detectar el mismo se debe obtener la derivada de la función de la imagen.

Una vez calculadas las derivadas de las direcciones “X” e “y”, implementamos una doble convolución en la que primero aplicamos un filtrado horizontal y, sobre el resultado aplicamos uno vertical. Como resultado obtenemos Cx e Cy. El objetivo es extraer características de gradiente en horizontal y vertical, simultáneamente.

Para finalizar el proceso, calculamos la magnitud del gradiente mediante el cálculo del valor máximo absoluto de las derivadas anteriores.

Línea 222:

Creamos el directorio donde se van a ir almacenando cada una de las imágenes que se han filtrado con el filtrado de SOBEL.

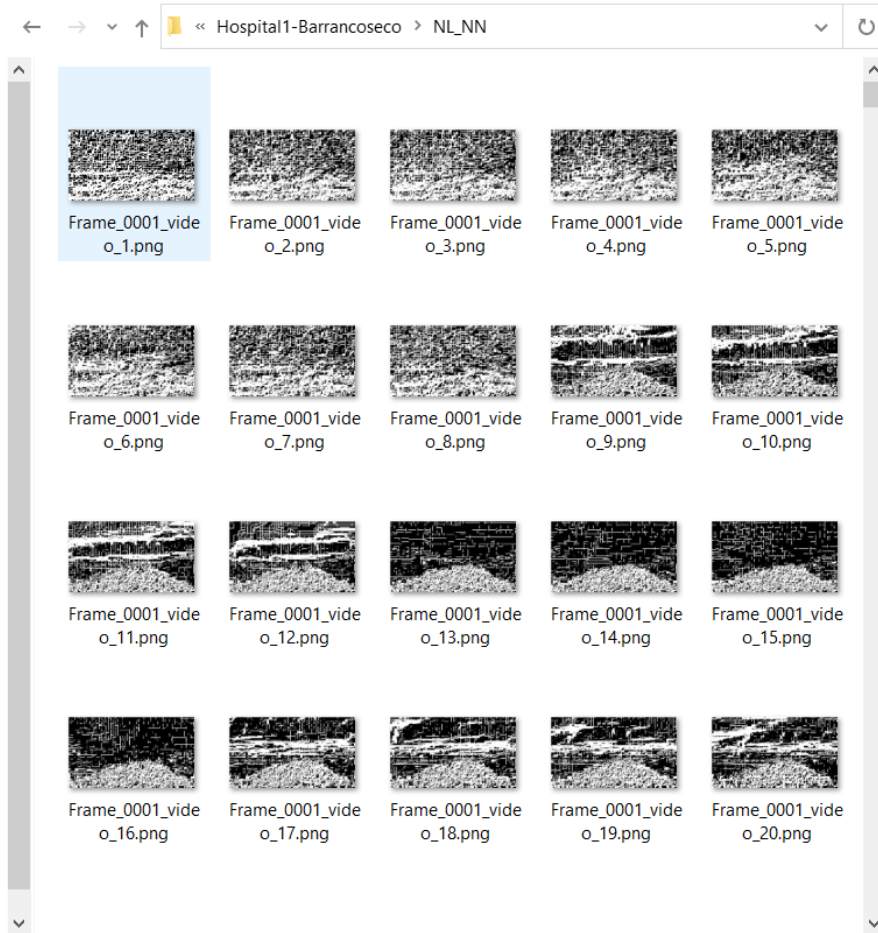


Imagen 169. Directorio donde se van almacenando las imágenes tras pasar por el filtrado de Sobel.

Línea 223:

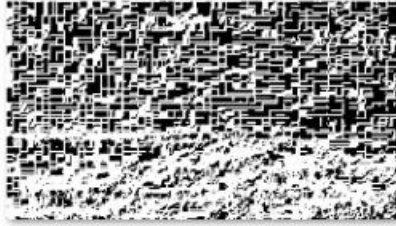
Hacemos una copia del directorio donde se van a ir almacenando las imágenes que se les ha aplicado el filtrado SOBEL.

Líneas 224-226:

Estructura condicional que verifica si este directorio existe con anterioridad o no. En el caso de que no exista este directorio lo creamos por primera vez.

Línea 228:

Le asignamos un nombre a la imagen antes de guardarla en su directorio correspondiente. Se le asigna un número a la imagen y el número del vídeo al que pertenece. Y se guarda en formato 'png'.



Frame_0001_video_1.png

Imagen 170. Nombre con el que se almacenan las imágenes tras pasar el filtro de Sobel.

Línea 229:

Mediante la función 'fullfile' de Matlab creamos la ruta donde se va a ir almacenando las imágenes que se le ha aplicado el filtro SOBEL.

Línea 230:

Se procede a guardar cada una de las imágenes anteriores.

Fragmento 13 del algoritmo:

```
232 - case {'c','C'}
233
234     % _____ CIELAB _____
235
236     %Pasamos la imagen a espacio de color lab (CIELAB)
237
238     Dir_CIELAB      = ['CIELAB' '\\' Camara '\\' DD(ii+2).name];
239     % Balance de blancos:
240     [archivoImagen]= WBLab4(Min_Frames,archivoImagen,Dir_CIELAB,frame,jj);
241
242     % Guardamos la imagen en RGB:
243     Dir_RGB         = ['RGB' '\\' Camara '\\' DD(ii+2).name];
244     DD_RGB         = dir(Dir_RGB);
245     if ( isempty(DD_RGB) )
246         mkdir(Dir_RGB);
247     end
248
249     outputBaseFileName = sprintf('Frame_%4.4d_video_%d.png', frame,jj);
250     outputFullFileName = fullfile(Dir_RGB, outputBaseFileName);
251     imwrite(archivoImagen, outputFullFileName, 'png');
```

Imagen 171. Fragmento 13 del algoritmo implementado en matlab.

Línea 238:

Creamos el directorio donde se van a ir almacenando cada una de las imágenes que se filtraran con el modo de color Cielab.

Línea 240:

Función 'WBLab4', esta función realiza varios pasos:

1. Se configura el filtro Gaussiano para posteriormente realizar el balance de blancos.
2. Pasamos las imágenes de RGB a Cielab.
3. Se realiza el balance de blancos regulando el contraste de cada componente: R, G y B.
4. Procedemos a guardar las imágenes en el directorio creado en la línea 238 del algoritmo principal.
5. Transformamos las imágenes a RGB con el modo de color Cielab y balance de blancos realizado.

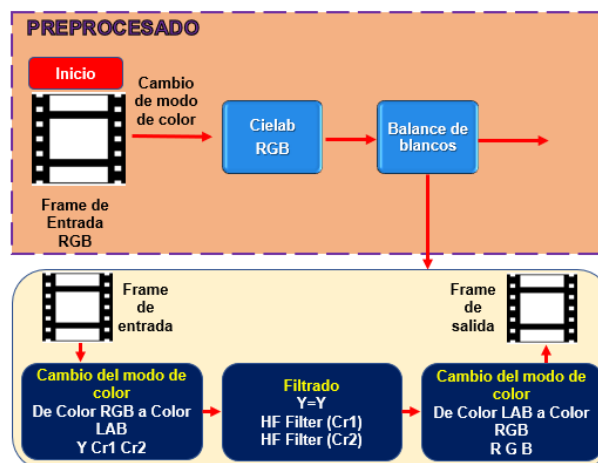


Imagen 172. Filtro de balance de blancos.

```

1  function [archivoImagen]= WBLab4(Min_Frames,archivoImagen, Dir_CIELAB,frame,jj)
2
3      % CONFIGURACIÓN DEL FILTRO GAUSSIANO
4  -   hsize=500;
5  -   corte=200;
6  -   W = fspecial('gaussian', hsize, corte);
7
8      % Pasamos de RGB a CIELAB
9
10 -   ImageLAB = rgb2lab(archivoImagen);
11
12      % Contraste de R
13 -   ImageLAB2(:, :,1)=ImageLAB(:, :,1);
14      % Contraste de G
15 -   ImageLAB2(:, :,2)=ImageLAB(:, :,2)-filter2(W, ImageLAB(:, :,2));
16      % Contraste de B
17 -   ImageLAB2(:, :,3)=ImageLAB(:, :,3)-filter2(W, ImageLAB(:, :,3));
18
19
20      % Guardamos la imagen en CIELAB:
21 -   DD_CIELAB      = dir(Dir_CIELAB);
22 -   if ( isempty(DD_CIELAB) )
23 -       mkdir(Dir_CIELAB);
24 -   end
25
26 -   outputBaseFileName = sprintf('Frame_%4.4d_video_%d.png', frame,jj);
27 -   outputFullFileName = fullfile(Dir_CIELAB, outputBaseFileName);
28 -   imwrite(archivoImagen, outputFullFileName, 'png');
29
30      % Pasamos de CIELAB a RGB
31 -   archivoImagen = lab2rgb(ImageLAB2);
32
33
34 - end

```

Imagen 173. Función de balance de blancos.

Líneas 242-251:

Realizamos el mismo procedimiento descrito anteriormente para la creación del directorio donde se van a ir almacenar las imágenes anteriores y se procede a su almacenamiento.

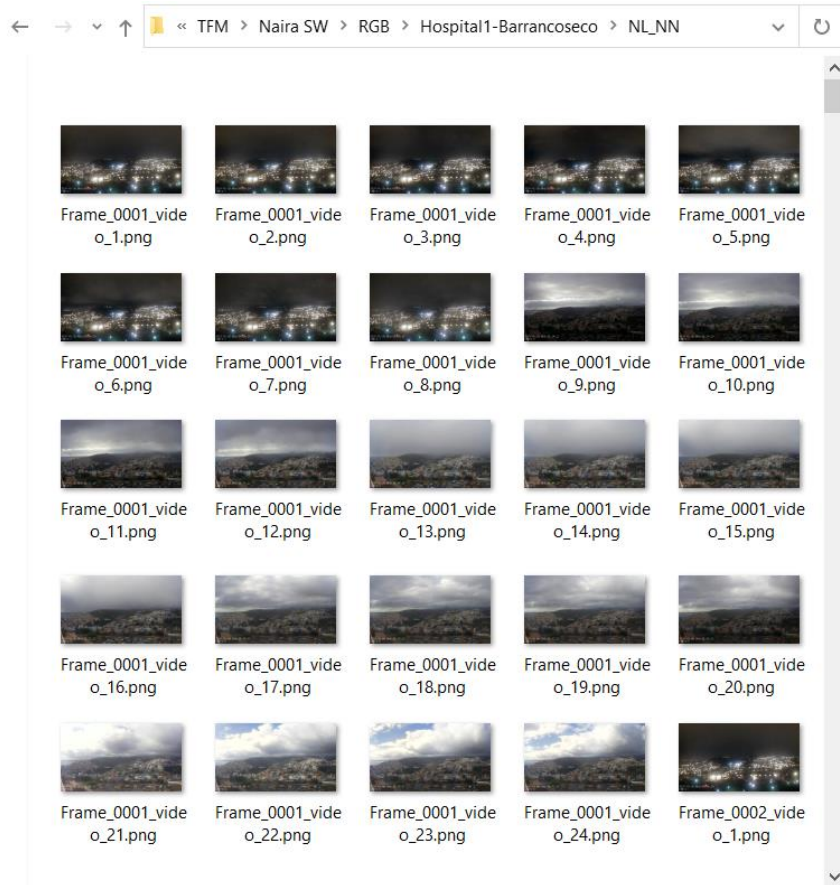


Imagen 174. Directorio del balance de blancos.

Fragmento 14 del algoritmo:

```
253 % Selecciono una sección de imagen de CIELAB:
254 I = imread(outputFullFileName, 'png');
255 [fil,col,prof] = size(I);
256
257 switch Camara
258     case 'Tafiral-Informatica'
259         % Segmento n° 8:
260         I2 = imcrop(I,[(col-1)/3 ((fil-1)/3)*2 ((col-1)/3 (fil-1)/3]);
261     case 'Tafira3-FOSCAM'
262         % Segmento n° 8:
263         I2 = imcrop(I,[(col-1)/3 ((fil-1)/3)*2 ((col-1)/3 (fil-1)/3]);
264     case 'Hospital1-Barrancoseco'
265         % Segmento n° 4:
266         I2 = imcrop(I,[0 (fil-1)/3 ((col-1)/3 (fil-1)/3]);
267     case 'Hospital2-Catedral'
268         % Segmento n° 9:
269         I2 = imcrop(I,[(col-1)/3)*2 ((fil-1)/3)*2 ((col-1)/3 (fil-1)/3]);
270     case 'Hospital3-Isleta'
271         % Segmento n° 9:
272         I2 = imcrop(I,[(col-1)/3)*2 ((fil-1)/3)*2 ((col-1)/3 (fil-1)/3]);
273     case 'Hospital4-Escaleritas'
274         % Segmento n° 8:
275         I2 = imcrop(I,[(col-1)/3 ((fil-1)/3)*2 ((col-1)/3 (fil-1)/3]);
276     case 'Tafira2-MontanaArucas'
277         % Segmento n° 5:
278         I2 = imcrop(I,[(col-1)/3 (fil-1)/3 ((col-1)/3 (fil-1)/3]);
279     case 'Tafira4-FOSCAM'
280         % Segmento n° 4:
281         I2 = imcrop(I,[0 (fil-1)/3 ((col-1)/3 (fil-1)/3]);
282     otherwise
283         %si es otra cámara se coje por defecto segmento n°5
284         I2 = imcrop(I,[(col-1)/3 (fil-1)/3 ((col-1)/3 (fil-1)/3]);
285 end
```

Imagen 175. Fragmento 14 del algoritmo implementado en matlab.

Línea 254:

En esta línea se va leyendo cada una de las imágenes almacenadas en la carpeta de RGB.

Línea 255:

Calculamos el tamaño de la imagen seleccionada anteriormente.

Líneas 257-285:

Estructura condicionante, dependiendo de la cámara se procede a seleccionar una región de interés, tal y como se explicó anteriormente.

Fragmento 15 del algoritmo:

```
286
287     % Guardo imagen de RGB segmentada/recortada:
288
289     Dir_RGB2      = ['Segmentación RGB' '\\' Camara '\\' DD(ii+2).name];
290     DD_RGB2      = dir(Dir_RGB2);
291     if ( isempty(DD_RGB2) )
292         mkdir(Dir_RGB2);
293     end
294     outputBaseFileName = sprintf('Frame_%4.4d_video_%d.png', frame,jj);
295     outputFullFileName = fullfile(Dir_RGB2, outputBaseFileName);
296     imwrite(I2, outputFullFileName, 'png');
```

Imagen 176. Fragmento 14 del algoritmo implementado en matlab.

Línea 289:

Creamos el directorio donde se van a ir almacenando cada una de las imágenes que se han filtrado con el modo de color de Cielab una vez se ha seleccionado la región de interés.

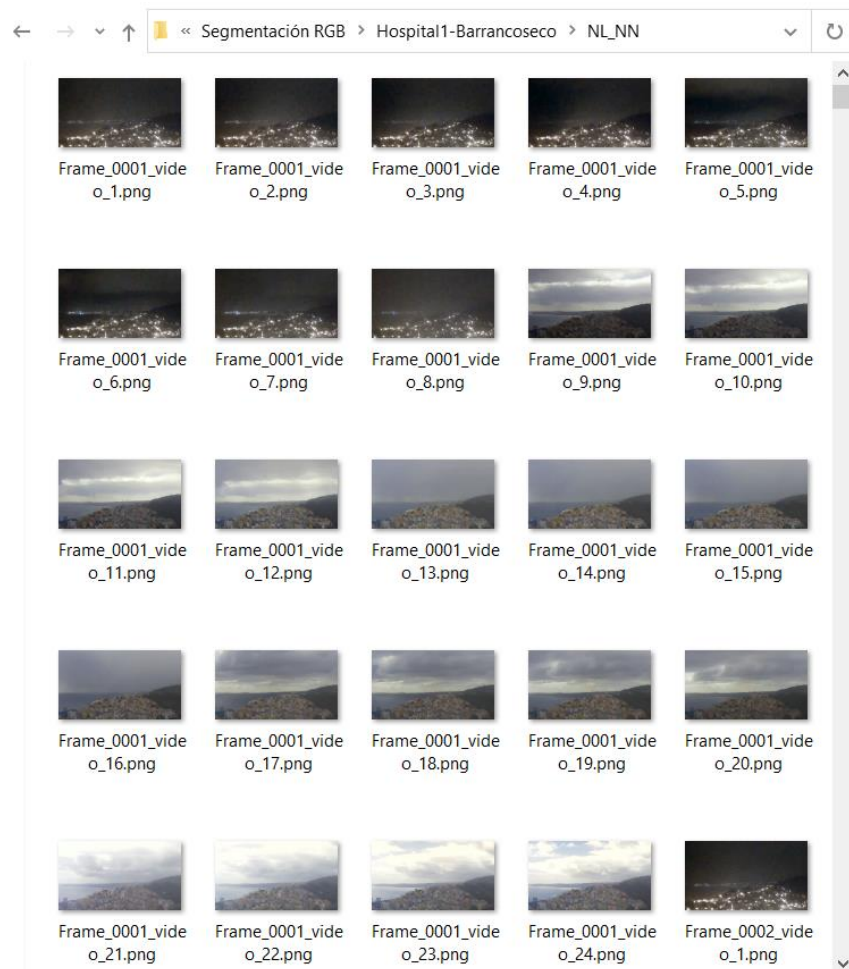


Imagen 177. Directorio tras el filtrado de modo de color Cielab.

Línea 290:

Hacemos una copia del directorio donde se van a ir almacenando las imágenes en modo de color Cielab con la región de interés seleccionada.

Líneas 291-293:

Estructura condicional que verifica si este directorio existe con anterioridad o no. En el caso de que no exista este directorio lo creamos por primera vez.

Línea 294:

Le asignamos un nombre a la imagen antes de guardarla en su directorio correspondiente. Se le asigna un número a la imagen y el número del vídeo al que pertenece. Y se guarda en formato 'png'.



Frame_0001_video_13.png

Imagen 178. Nombre de cada imagen tras el filtrado de modo de color Cielab.

Línea 295:

Mediante la función 'fullfile' de Matlab creamos la ruta donde se va a ir almacenando las imágenes de modo de color Cielab con la región de interés seleccionada.

Línea 296:

Se procede a guardar cada una de las imágenes en modo de color de Cielab con la región de interés seleccionada en la ruta generada en la línea 295.

Fragmento 16 del algoritmo:

```
298 % Filtrado de Sobel-----
299 if (Filt_Sobel==1)
300     [Is,Js,Ks] = size(I2);
301     I2_b      = I2;
302     for ss=1:Ks
303         if (Filt_Sobel==1)
304             Cx = conv2(I2_b(:,:,ss),Mx);
305             Cy = conv2(I2_b(:,:,ss),My);
306             I2(:,:,ss) = max(Cx(Borde_Sobel+1:end-Borde_Sobel,Borde_Sobel+1:end-Borde_Sobel),Cy(Borde_Sobel+1:end-Borde_Sobel,Borde_Sobel+1:end-Borde_Sobel));
307         end
308     end
309 end
310 end
311
312 % Guardo imagen del filtro Sobel:
313 Dir_RGB3 = ['segmentación RGB y SOBEL' '\\' Camara '\\' DD(ii+2).name];
314 DD_RGB3 = dir(Dir_RGB3);
315 if ( isempty(DD_RGB3) )
316     mkdir(Dir_RGB3);
317 end
318 outputBaseFileName = sprintf('Frame_%4.4d_video_%4d.png', frame, jj);
319 outputFullFileName = fullfile(Dir_RGB3, outputBaseFileName);
320 imwrite(I2, outputFullFileName, 'png');
```

Imagen 179. Fragmento 16 del algoritmo implementado en matlab.

Líneas 299-308:

A las imágenes segmentadas se le aplica el filtrado de SOBEL, tal como ya se ha explicado en las líneas 213-219.

Línea 313:

Creamos el directorio donde se van a ir almacenando cada una de las imágenes que se han filtrado con el filtrado de SOBEL.

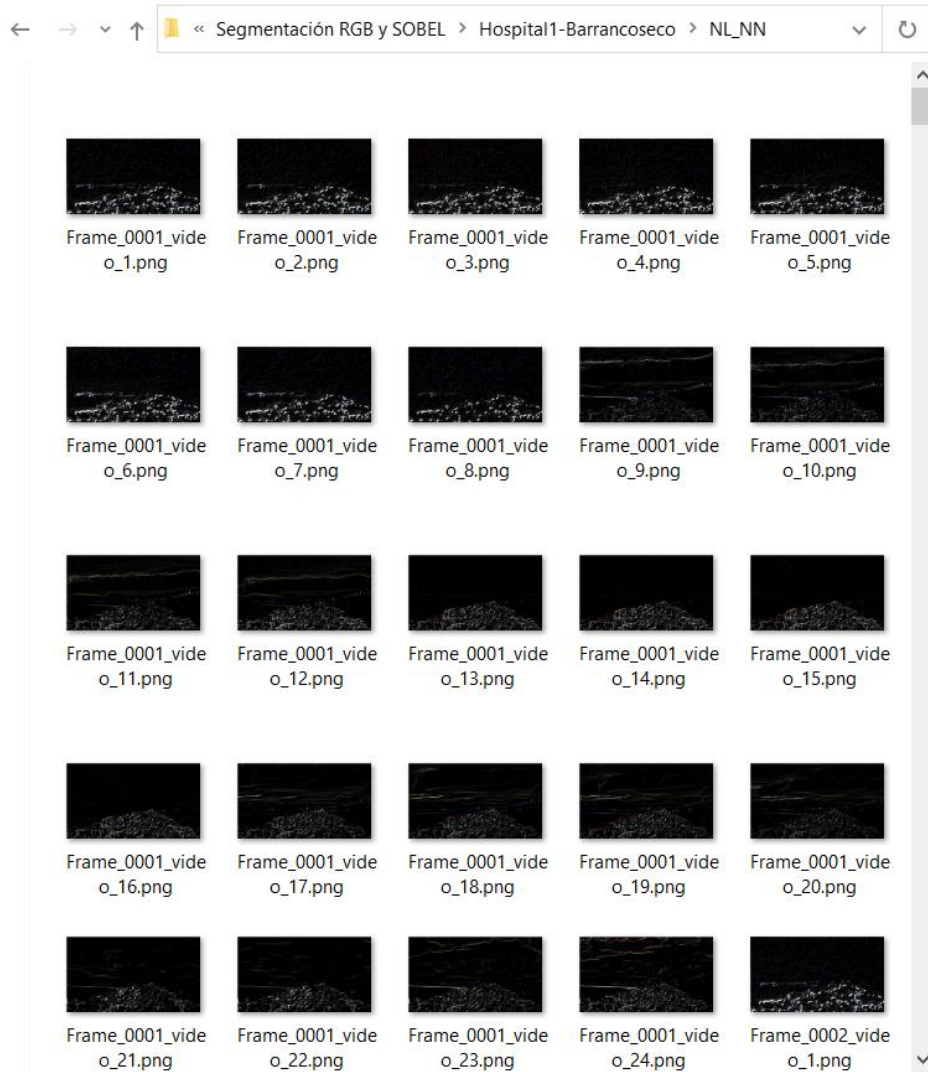


Imagen 180. Directorio después de que las imagen se filtren con el filtrado Sobel.

Línea 314:

Hacemos una copia del directorio donde se van a ir almacenando las imágenes que se les ha aplicado el filtrado SOBEL.

Líneas 315-317:

Estructura condicional que verifica si este directorio existe con anterioridad o no. En el caso de que no exista este directorio lo creamos por primera vez.

Línea 318:

Le asignamos un nombre a la imagen antes de guardarla en su directorio correspondiente. Se le asigna un número a la imagen y el número del vídeo al que pertenece. Y se guarda en formato 'png'.



Frame_0001_video_5.png

Imagen 181. Nombre de la imagen tras pasar por el filtro Sobel.

Línea 319:

Mediante la función 'fullfile' de Matlab creamos la ruta donde se va a ir almacenando las imágenes que se le ha aplicado el filtro SOBEL.

Línea 320:

Se procede a guardar cada una de las imágenes anteriores.

Fragmento 17 del algoritmo:

```
332 - if (Filt_Sobel == 1) %si filtro SOBER
333 -     switch opcion
334 -         case {'L','l'}
335 -             cad='Segmentación Luminancia y SOBEL';
336 -         case {'C','c'}
337 -             cad='Segmentación RGB y SOBEL';
338 -         end
339 -     else %No filtro SOBER
340 -         switch opcion
341 -             case {'L','l'}
342 -                 cad='Segmentación Luminancia';
343 -             case {'C','c'}
344 -                 cad='Segmentación RGB';
345 -             end
346 -         end
347 -
348 -         inputFolderVideo = [pwd '\\' cad '\\' Camara];
```

Imagen 182. Fragmento 17 del algoritmo implementado en matlab.

Líneas 332-348:

En estas líneas mediante un condicional recuperamos la ruta del directorio donde se han almacenado las imágenes que se les ha aplicado el último filtrado de la parte de pre-procesado, es decir, el filtrado SOBEL. Y almacenamos el directorio en la variable 'inputFolderVideo'.

Fragmento 18 del algoritmo:

```
351 - digitDatasetPath = fullfile(inputFolderVideo,{'SL_NB','SL_NA','NL_NN'}); %disponemos de 3 clases
352 - imds = imageDatastore(digitDatasetPath,'LabelSource','foldernames','FileExtensions',{'png'});
353
354 - % Calculate the number of images in each category.
355 - img = readimage(imds,1);
356 - [n,m,w] = size(img);
357 - Train_percent = 0.7;
358 - [imdsTrain,imdsValidation] = splitEachLabel(imds,Train_percent,'randomize');
```

Imagen 183. Fragmento 18 del algoritmo implementado en matlab.

Línea 351:

Con esta función de Matlab abrimos 3 directorios a modo de ráfaga, que representan las 3 clases del aprendizaje supervisado de la red neuronal:

- **NL_NN:** No hay datos meteorológicos reales de lluvia y el nivel de precipitación es nulo.
- **SL_NB:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es bajo.
- **SL_NA:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es alto.

Línea 352:

Con esta función podemos leer y almacenar las imágenes de los directorios anteriores a modo de ráfaga. En cuanto a sus parámetros de entrada:

- **'digitDatasetPath':** Representa la localización donde se encuentran almacenadas las imágenes leídas a modo de ráfaga anteriormente.
- **'LabelSource':** Este parámetro indica la fuente de la etiqueta para las imágenes. Para asignar una fuente por defecto se ha introducido seguidamente 'foldernames'.

- **'FileExtensions'**: Este parámetro indica la extensión de las imágenes. Por ello a continuación se ha introducido la extensión que poseen dichas imágenes: '.png'.

Y se almacenan en la variable de salida imds.

Línea 355:

Esta función lee la primera imagen del almacén de imágenes imds y la almacena en la variable img. En cuanto a sus parámetros de entrada son:

- **Nombre del almacén de las imágenes 'Imds'**: nombre del almacén de imágenes.
- **Número de la imagen que se desea leer**: Se debe de indicar con un valor numérico, en nuestro caso queremos leer una imagen cualquiera, ya que nos interesa conocer su tamaño posterior. Como los tamaños de todas las imágenes son iguales se ha procedido a la lectura de la primera imagen, por ello, se la introducido el valor numérico 1.

Por último, esta imagen leída se almacena en la variable de salida imd.

Línea 356:

Con esta función de Matlab calculamos el ancho, alto y profundidad de la imagen leída anteriormente.

Línea 357:

Se impone un porcentaje de entrenamiento al 70% y, por lo tanto, se dispone por defecto de un 30% para la validación.

Línea 358:

Se divide aleatoriamente las imágenes en dos grupos, uno para el entrenamiento y el otro para la validación. Se ha utilizado el 70% de las imágenes para el entrenamiento y el resto para la validación.

Fragmento 19 del algoritmo:

```
360 % Define the convolutional neural network architecture.
361 - layers = [
362     imageInputLayer([n m w])
363     convolution2dLayer(3,8,'Padding','same')
364     batchNormalizationLayer
365     reluLayer
366
367     maxPooling2dLayer(2,'Stride',2)
368
369     convolution2dLayer(3,16,'Padding','same')
370     batchNormalizationLayer
371     reluLayer
372
373     maxPooling2dLayer(2,'Stride',2)
374
375     convolution2dLayer(3,32,'Padding','same')
376     batchNormalizationLayer
377     reluLayer
378
379     fullyConnectedLayer(3) % Tres clases de salida
380     softmaxLayer
381     classificationLayer];
```

Imagen 184. Fragmento 19 del algoritmo implementado en matlab.

Líneas 361-377:

Tal y como se explicó en el capítulo 6, las capas ocultas o intermedias de la red neuronal convolucional se basan en capas formadas por neuronas que se encargan de realizar todo el aprendizaje del algoritmo de la red neuronal. Según el algoritmo de aprendizaje, se divide en dos tipos de capas:

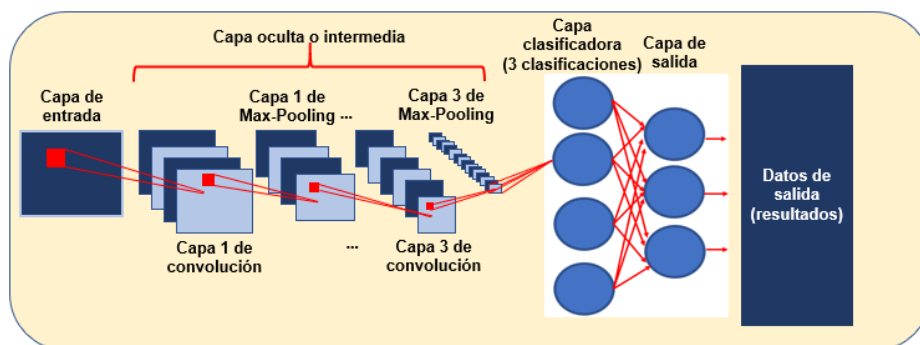


Imagen 185. Red Neuronal Convolucional.

- **Capa convolucional:** Esta capa realiza las operaciones de productos y sumas para generar un mapa de características. Cada una de estas características corresponde con una ubicación del filtro de la imagen

original. Los parámetros de entrada que se han introducido son los siguientes:

- **Tamaño del filtro:** Las imágenes disponen de 3 dimensiones, estas son alto, ancho y profundidad.
- **Número de filtros:** Se han seleccionado 8 filtros.
- **Tamaño de las imágenes de salida:** Para que las imágenes de entrada y salida tengan el mismo tamaño se recurre a rellenar con ceros hasta alcanzar el mismo tamaño. Por ello, se añade la variable 'Padding', que representa que introducimos relleno de borde de entrada y, posteriormente la variable 'same' que indica que queremos establecer el mismo tamaño de las imágenes de entrada con respecto de las imágenes de salida.

Una vez realizada la convolución de las imágenes se procede a normalizar las imágenes para obtener una velocidad mayor en el proceso de entrenamiento. Para ello, se usa la función 'batchNormalizationLayer'.

Posteriormente, mediante la función de Matlab 'reluLayer', se le aplica a cada elemento de entrada una operación para establecer un nivel umbral:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Imagen 186. Función 'reluLayer'.

- **Capa Max-Pooling:** Es la capa que realiza la reducción de los datos, es decir, reduce el ancho y el alto de las imágenes obtenidas de la capa convolucional, pero la profundidad la mantiene tal cual la recibió de la capa convolucional. Se basa en disminuir la cantidad de datos que presentan características comunes. Max-Pooling calcula el máximo valor de una región determinada como se observa en la siguiente imagen.

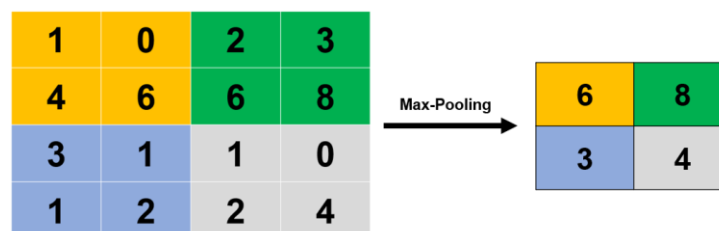


Imagen 187. Max-Pooling.

Los parámetros de entrada son:

- **Dimensiones de las regiones de la agrupación**
- **Tamaño de paso para atravesar la entrada:** Al introducir la variable 'Stride' estamos indicando que deseamos el mismo valor para las dimensiones del tamaño vertical y horizontal.
- **Tamaño del relleno:** al introducir la variable 'same' estamos indicando que deseamos que los elementos posean el mismo tamaño de relleno.

Línea 379:

Esta función indica el número de clases de salida de la red neuronal convolucional. En este TFM son 3 clases de salida.

Línea 380:

Esta función crea la capa previa a la capa de salida, esta capa debe de contener el mismo número de nodos que la capa de salida o de resultados. A demás, también calcula una probabilidad para cada clase disponible.

Línea 381:

Función de Matlab que crea la capa de clasificación.

Fragmento 20 del algoritmo:

```
383 | % Specify Training Options
384 | options = trainingOptions('sgdm', ...
385 |     'InitialLearnRate',0.01, ...
386 |     'MaxEpochs',6, ...
387 |     'Shuffle','every-epoch', ...
388 |     'ValidationData',imdsValidation, ...
389 |     'ValidationFrequency',10, ...
390 |     'Verbose',false, ...
391 |     'Plots','training-progress','MiniBatchSize',32);
```

Imagen 188. Fragmento 20 del algoritmo implementado en matlab.

Líneas 383-391:

Con esta función de Matlab se especifican todas las opciones del entrenamiento de la red neuronal convolucional.

Las opciones propuestas en este TFM son:

- **'sgdm'**: Para optimizar el descenso de gradiente estocástico con impulso.
- **'InitialLearnRate'**: Tasa de aprendizaje inicial empleada para el entrenamiento. Su valor depende del optimizador empleado anteriormente:
 - **'sgdm'**: su valor predeterminado es 0.01.
 - **'rmsprop' o 'adam'**: su valor predeterminado es 0.001.

Si la tasa de aprendizaje es baja el entrenamiento puede conllevar mucho tiempo en su desarrollo.

- **'MaxEpochs'**: Número máximo de épocas para el entrenamiento. Una época es el paso completo del algoritmo de entrenamiento sobre todo el conjunto de entrenamiento. En este TFM se ha seleccionado 6 épocas, porque tras varios experimentos hemos observado que los resultados obtenidos se estabilizan y se mantienen con los mismos valores de resultados a partir de la tercera época.
- **'shuffle'**: Opción para mezclar los datos. Se dispone de tres posibilidades:
 - **'once'**: mezcla los datos de entrenamiento y validación una vez antes del entrenamiento.
 - **'never'**: no mezcla los datos.
 - **'every-epoch'**: mezcla los datos de entrenamiento antes de cada período de entrenamiento y mezcla los datos de validación antes de cada validación de red.

La opción elegida es 'every-epoch' para que nos ofrezca mayor aleatoriedad en cada período.

- **'ValidationData'**: Datos o almacenamiento de datos de imagen que se utilizarán para la validación durante el entrenamiento.
- **'ValidationFrequency'**: Frecuencia de validación de la red en cuanto al número de iteraciones.

En este TFM se ha seleccionado un valor de frecuencia de validación de 10 para no ralentizar el entrenamiento.

- **'verbose'**: Indicador para mostrar información sobre el progreso del entrenamiento en la ventana de comandos. Existen dos opciones:
 - **'false'**: No se muestra la información.
 - **'true'**: Si se muestra la información.

En este TFM se ha seleccionado la segunda opción para agilizar el tiempo empleado en el entrenamiento.

- **'Plots'**: Para mostrar los gráficos durante el entrenamiento de la red neuronal convolucional. Existen dos opciones:
 - **'none'**: no muestra gráficos durante el entrenamiento.
 - **'training-progress'**: si muestra gráficos en el progreso del entrenamiento.

En este TFM se ha seleccionado la segunda opción para poder observar los resultados de los experimentos realizados en el proceso de entrenamiento.

- **'MiniBatchSize'**: Se basa en un número entero positivo que representa un subconjunto del conjunto total de entrenamiento que se usa para evaluar el gradiente de la función de pérdidas y actualizar los pesos.

Dicho de otra manera, si hay muchas imágenes las toma en grupos especificados en esta variable. En cada iteración de entrenamiento puede coger diferentes imágenes de las que hay para entrenar, pero siempre esta cantidad de imágenes indicada en esta variable.

En este TFM se ha seleccionado un valor de 32, pero tiene la opción de realizar grupos de hasta 256. Este valor se obtuvo realizando pruebas

hasta que los programas funcionaban bien, esto es, entrena bien y los test dan buenos resultados en las métricas de calidad.

Fragmento 21 del algoritmo:

```
393 | % Train Network Using Training Data
394 |
395 - | net          = trainNetwork(imdsTrain, layers, options);
396 |
397 | % Classify Validation Images and Compute Accuracy
398 - | YPred       = classify(net, imdsValidation);
399 - | YValidation  = imdsValidation.Labels;
400 |
401 - | accuracy    = sum(YPred == YValidation)/numel(YValidation);
402 - | fprintf('El accuracy: %.4f \n', accuracy);
```

Imagen 189. Fragmento 21 del algoritmo implementado en matlab.

Línea 395:

Función de Matlab para realizar el entrenamiento con los datos especificados en la línea 384.

Líneas 397- 402:

Se procede a la clasificación de las imágenes de validación y se calcula la precisión.

El parámetro de la precisión representa una proporción entre el número de eventos detectados como casos positivos de lluvia entre el número de eventos totales reales u observados negativos de lluvia (datos meteorológicos reales de no lluvia). Y se calcula con la fórmula siguiente.

$$\text{Precisión} = \frac{n^{\circ} \text{ de eventos detectados negativos}}{n^{\circ} \text{ de eventos totales reales negativos}} = \frac{TP}{TP+FP}$$

Ecuación 17. Fórmula para el cálculo de la precisión.

Fragmento 22 del algoritmo:

```
406 - TP           = 0; % True positive,   rain detected as rain
407 - FP           = 0; % False positive,  no-rain detected as rain
408 - TN           = 0; % True negative,   no-rain detected as no-rain
409 - FN           = 0; % False negative,  rain detected as no-rain
410 - Itest        = length(YValidation);
411 - for jj=1:Itest
412 -     if        ( strcmp(char(YPred(jj)), 'NL_NN') && strcmp(char(YValidation(jj)), 'NL_NN') )
413 -         TN = TN+1;
414 -     elseif    ( strcmp(char(YPred(jj)), 'SL_NB') && strcmp(char(YValidation(jj)), 'NL_NN') )
415 -         FP = FP+1;
416 -     elseif    ( strcmp(char(YPred(jj)), 'SL_NA') && strcmp(char(YValidation(jj)), 'NL_NN') )
417 -         FP = FP+1;
418 -     elseif    ( strcmp(char(YPred(jj)), 'NL_NN') && strcmp(char(YValidation(jj)), 'SL_NB') )
419 -         FN = FN+1;
420 -     elseif    ( strcmp(char(YPred(jj)), 'SL_NB') && strcmp(char(YValidation(jj)), 'SL_NB') )
421 -         TP = TP+1;
422 -     elseif    ( strcmp(char(YPred(jj)), 'SL_NA') && strcmp(char(YValidation(jj)), 'SL_NB') )
423 -         TP = TP+1;
424 -     elseif    ( strcmp(char(YPred(jj)), 'NL_NN') && strcmp(char(YValidation(jj)), 'SL_NA') )
425 -         FN = FN+1;
426 -     elseif    ( strcmp(char(YPred(jj)), 'SL_NB') && strcmp(char(YValidation(jj)), 'SL_NA') )
427 -         TP = TP+1;
428 -     elseif    ( strcmp(char(YPred(jj)), 'SL_NA') && strcmp(char(YValidation(jj)), 'SL_NA') )
429 -         TP = TP+1;
430 -     end
431 - end
```

Imagen 190. Fragmento 22 del algoritmo implementado en matlab.

Líneas 406 - 431:

Como ya se explicó en el capítulo 6, para definir los parámetros de calidad para la detección es necesario establecer, inicialmente, una matriz de confusión de 2x2. Dónde los parámetros usados en dicha matriz son los siguientes:

- **TP:** en inglés “True Positive” o en castellano “Verdadero Positivo”. Se detecta lluvia y hay datos meteorológicos reales de lluvia.
- **FN:** en inglés “False Negative” o en castellano “Falso Positivo”. Se detecta no lluvia y hay datos meteorológicos reales de lluvia.
- **TN:** en inglés “True Negative” o en castellano “Verdadero Negativo”. Se detecta no lluvia y no hay datos meteorológicos reales de lluvia.
- **FP:** en inglés “False Positive” o en castellano “Falso Negativo”. Se detecta lluvia y no hay datos meteorológicos reales de lluvia.

Recordemos que disponemos de 2 clases en el aprendizaje supervisado de la red neuronal para el proceso de detección:

- **NL:** No hay datos meteorológicos reales de lluvia.
- **SL:** Si hay datos meteorológicos reales de lluvia.

| | | Lluvia detectada o su predicción | |
|-------------------------|----|----------------------------------|----|
| | | NL | SL |
| Lluvia real u observada | NL | TN | FP |
| | SL | FN | TP |

Tabla 16. Matriz binaria para la detección de la precipitación.

Fragmento 23 del algoritmo:

```

434 % MATRIZ DE CONFUSIÓN 3x3 cuantificación
435 - Matriz_Confusión = zeros(3,3);
436 - for jj=1:Itest
437 -     if (strcmp(char(YPred(jj)), 'NL_NN')) && (strcmp(char(YValidation(jj)), 'NL_NN')) )
438 -         Matriz_Confusión(1,1) = Matriz_Confusión(1,1)+1;
439 -     elseif (strcmp(char(YPred(jj)), 'SL_NB')) && (strcmp(char(YValidation(jj)), 'NL_NN')) )
440 -         Matriz_Confusión(1,2) = Matriz_Confusión(1,2)+1;
441 -     elseif (strcmp(char(YPred(jj)), 'SL_NA')) && (strcmp(char(YValidation(jj)), 'NL_NN')) )
442 -         Matriz_Confusión(1,3) = Matriz_Confusión(1,3)+1;
443 -     elseif (strcmp(char(YPred(jj)), 'NL_NN')) && (strcmp(char(YValidation(jj)), 'SL_NB')) )
444 -         Matriz_Confusión(2,1) = Matriz_Confusión(2,1)+1;
445 -     elseif (strcmp(char(YPred(jj)), 'SL_NB')) && (strcmp(char(YValidation(jj)), 'SL_NB')) )
446 -         Matriz_Confusión(2,2) = Matriz_Confusión(2,2)+1;
447 -     elseif (strcmp(char(YPred(jj)), 'SL_NA')) && (strcmp(char(YValidation(jj)), 'SL_NB')) )
448 -         Matriz_Confusión(2,3) = Matriz_Confusión(2,3)+1;
449 -     elseif (strcmp(char(YPred(jj)), 'NL_NN')) && (strcmp(char(YValidation(jj)), 'SL_NA')) )
450 -         Matriz_Confusión(3,1) = Matriz_Confusión(3,1)+1;
451 -     elseif (strcmp(char(YPred(jj)), 'SL_NB')) && (strcmp(char(YValidation(jj)), 'SL_NA')) )
452 -         Matriz_Confusión(3,2) = Matriz_Confusión(3,2)+1;
453 -     elseif (strcmp(char(YPred(jj)), 'SL_NA')) && (strcmp(char(YValidation(jj)), 'SL_NA')) )
454 -         Matriz_Confusión(3,3) = Matriz_Confusión(3,3)+1;
455 -     end
456 - end
457 % FIN - MATRIZ DE CONFUSIÓN 3x3 cuantificación
458
459 % Excel matriz de confusión 3x3 cuantificación
460 - xlswrite(filename2,C,hoja2,Rango3);
461 - xlswrite(filename2,D,hoja2,Rango4);
462 - xlswrite(filename2,Matriz_Confusión(1,1),hoja2,'B2:B2');
463 - xlswrite(filename2,Matriz_Confusión(1,2),hoja2,'C2:C2');
464 - xlswrite(filename2,Matriz_Confusión(1,3),hoja2,'D2:D2');
465 - xlswrite(filename2,Matriz_Confusión(2,1),hoja2,'B3:B3');
466 - xlswrite(filename2,Matriz_Confusión(2,2),hoja2,'C3:C3');
467 - xlswrite(filename2,Matriz_Confusión(2,3),hoja2,'D3:D3');
468 - xlswrite(filename2,Matriz_Confusión(3,1),hoja2,'B4:B4');
469 - xlswrite(filename2,Matriz_Confusión(3,2),hoja2,'C4:C4');
470 - xlswrite(filename2,Matriz_Confusión(3,3),hoja2,'D4:D4');
471 - hoja2=hoja2+1;
472 % FIN Excel matriz de confusión 3x3 cuantificación

```

Imagen 191. Fragmento 23 del algoritmo implementado en matlab.

Líneas 434-457:

Como ya se explicó en el capítulo 6, para medir los parámetros de calidad para la cuantificación es necesario ampliar la matriz de confusión a un tamaño de 3x3.

En este momento disponemos de 3 clases en el aprendizaje supervisado de la red neuronal:

- **NL_NN:** No hay datos meteorológicos reales de lluvia y el nivel de precipitación es nulo.
- **SL_NB:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es bajo.
- **SL_NA:** Si hay datos meteorológicos reales de lluvia y el nivel de precipitación es alto.

En la tabla siguiente se muestra los resultados obtenidos en la detección de la precipitación, se trata de una matriz de confusión de tamaño 3x3. Las filas de dicha matriz representan la lluvia real u observada, mientras que las columnas representan la lluvia detectada o su predicción.

| | | Lluvia detectada o su predicción | | |
|-------------------------|-------|----------------------------------|-------|-------|
| | | NL_NN | SL_NB | SL_NA |
| Lluvia real u observada | NL_NN | TN | FP | FP |
| | SL_NB | FN | TP | TP |
| | SL_NA | FN | TP | TP |

Tabla 17. Matriz de confusión.

Líneas 459-472:

Con estas líneas guardamos los datos obtenidos de los parámetros de cuantificación en un archivo Excel.

| | A | B | C | D |
|---|-------|-------|-------|-------|
| 1 | | NL_NN | SL_NB | SL_NA |
| 2 | NL_NN | 215 | 0 | 0 |
| 3 | SL_NB | 0 | 216 | 0 |
| 4 | SL_NA | 0 | 9 | 0 |

Imagen 192. Excel de la matriz de confusión.

Fragmento 24 del algoritmo:

```

475 - Scores(cc,1,1) = 100*(TN + TP)/(TN+TP+FN+FP); % Accuracy (Exactitud)
476 - Scores(cc,1,2) = 100*TP/(TP + FN); % Sensitivity (Sensibilidad)
477 - Scores(cc,2,1) = 100*TN/(TN + FP); % Specificity (Especificidad)
478 - Scores(cc,2,2) = 100*TP/(TP + FP); % Precision (Precisión)
479
480 - Scores_Copia(1) = Scores(cc,1,1);
481 - Scores_Copia(2) = Scores(cc,1,2);
482 - Scores_Copia(3) = Scores(cc,2,1);
483 - Scores_Copia(4) = Scores(cc,2,2);
484
485 - ScoresC(cc,1,1) = TN; %verdaderos negativos
486 - ScoresC(cc,1,2) = FP; %falsos positivos
487 - ScoresC(cc,2,1) = FN; %falsos negativos
488 - ScoresC(cc,2,2) = TP; %verdaderos positivos
489
490 - ScoresC_Copia(1)=ScoresC(cc,1,1);
491 - ScoresC_Copia(2)=ScoresC(cc,1,2);
492 - ScoresC_Copia(3)=ScoresC(cc,2,1);
493 - ScoresC_Copia(4)=ScoresC(cc,2,2);
494
495 %Guardamos los resultados en un fichero excel:
496
497
498 - Rango1 = strcat('A',num2str(filal),' ','D',num2str(filal));
499 - Rango2 = strcat('A',num2str(fila2),' ','D',num2str(fila2));
500 - xlswrite(filename,Scores_Copia,hoja1,Rango1)
501 - xlswrite(filename,ScoresC_Copia,hoja1,Rango2)
502 - filal=filal+1;
503 - fila2=fila2+1;

```

Imagen 193. Fragmento 24 del algoritmo implementado en matlab.

Líneas 475 - 503:

Con estas líneas guardamos los datos obtenidos de los parámetros de detección en un archivo Excel.

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,15 | 97,12 | 94,88 | 96,09 |
| 3 | 99,72 | 100,00 | 99,31 | 99,54 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 99,05 | 98,99 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 204 | 11 | 8 | 270 |
| 14 | 143 | 1 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 329 | 3 | 3 | 294 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 0 | 297 | 0 | 297 |

Imagen 194. Tabla con los parámetros calculados en las líneas anteriores.

Fragmento 25 del algoritmo:

```
520 -     f_exac = Scores_Copia(1)/100;
521 -     f_sen  = Scores_Copia(2)/100;
522 -     f_prec = Scores_Copia(3)/100;
523 -     f_esp  = Scores_Copia(4)/100;
524
525 -     N_exac = TP+TN+FP+FN;
526 -     N_sen  = TP+FN;
527 -     N_prec = TP+FP;
528 -     N_esp  = TN+FP;
529
530 -     [LS_exac,LI_exac,LS_sen,LI_sen,LS_esp,LI_esp,LS_prec,LI_prec] = NivelConfianza_MatrizBinaria (f_exac,N_exac,f_sen,N_sen,f_esp,N_esp,f_prec,N_prec);
531
532
533 -     % Excel Nivel de Confianza matriz binaria
534 -     vec_Nivel_Confianza=[LI_exac,f_exac,LS_exac,LI_sen,f_sen,LS_sen,LI_esp,f_esp,LS_esp,LI_prec,f_prec,LS_prec];
535 -     RangoNC2 = strcat('A',num2str(fila3),' ','I',num2str(fila3));
536 -     xlswrite(filename3,vec_Nivel_Confianza,hoja3,RangoNC2);
537 -     fila3=fila3+1;
```

Imagen 195. Fragmento 25 del algoritmo implementado en matlab.

Líneas 520 - 537:

Cálculo de los niveles de confianza de la matriz binaria o matriz de detección.

Posteriormente se exportan los resultados al fichero Excel creado en el inicio.

Fragmento 26 del algoritmo:

```
558 -     N_exac_NN = Matriz_Confesion(1,1) + Matriz_Confesion(1,2) + Matriz_Confesion(1,3);
559 -     N_exac_NB = Matriz_Confesion(2,1) + Matriz_Confesion(2,2) + Matriz_Confesion(2,3);
560 -     N_exac_NA = Matriz_Confesion(3,1) + Matriz_Confesion(3,2) + Matriz_Confesion(3,3);
561
562 -     f_exac_NN = Matriz_Confesion(1,1)/N_exac_NN;
563 -     f_exac_NB = Matriz_Confesion(2,2)/N_exac_NB;
564 -     f_exac_NA = Matriz_Confesion(3,3)/N_exac_NA;
565
566
567
568
569 -     exac_NN = (Matriz_Confesion(1,1)/N_exac_NN);
570 -     exac_NB = (Matriz_Confesion(2,2)/N_exac_NB);
571 -     exac_NA = (Matriz_Confesion(3,3)/N_exac_NA);
572
573 -     [LS_exac_NN,LI_exac_NN,LS_exac_NB,LI_exac_NB,LS_exac_NA,LI_exac_NA] = NivelConfianza_MatrizDeConfesion (f_exac_NN,N_exac_NN,f_exac_NB,N_exac_NB,f_exac_NA,N_exac_NA);
574
575
576 -     % Excel Nivel de Confianza matriz confusión
577 -     vec_Nivel_Confianza_MC=[LI_exac_NN,exac_NN,LS_exac_NN,LI_exac_NB,exac_NB,LS_exac_NB,LI_exac_NA,exac_NA,LS_exac_NA];
578 -     RangoNC2 = strcat('A',num2str(filaNCMC),' ','I',num2str(filaNCMC));
579 -     xlswrite(filenameNCMC,vec_Nivel_Confianza_MC,hojaNCMC,RangoNC2);
580 -     filaNCMC=filaNCMC+1;
```

Imagen 196. Fragmento 66 del algoritmo implementado en matlab.

Líneas 558 - 580:

Cálculo de los niveles de confianza de la matriz de confusión.

Posteriormente se exportan los resultados al fichero Excel creado en el inicio.

Anexo II

Experimentos sobre la profundidad de la red neuronal:

- Resultados usando el modo de color luminancia y se realizaron 9 iteraciones. En la Tabla 18 se muestra 9 iteraciones.

Tabla 18. Resultados de las 9 iteraciones con el modo de color luminancia.

| | A | B | C | D |
|-----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,15 | 97,12 | 94,88 | 96,09 |
| 3 | 99,72 | 100,00 | 99,31 | 99,54 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 99,05 | 98,99 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 204 | 11 | 8 | 270 |
| 14 | 143 | 1 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 329 | 3 | 3 | 294 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 0 | 297 | 0 | 297 |
| ... | | | | |
| | A | B | C | D |
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 97,16 | 98,56 | 95,35 | 96,48 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 99,82 | 100,00 | 99,67 | 99,59 |
| 5 | 100,00 | 100,00 | 100,00 | 100,00 |
| 6 | 99,36 | 99,33 | 99,40 | 99,33 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 99,83 | 100,00 | 99,66 | 99,66 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 205 | 10 | 4 | 274 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 304 | 1 | 0 | 242 |
| 16 | 232 | 0 | 0 | 223 |
| 17 | 330 | 2 | 2 | 295 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 296 | 1 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 97,77 | 97,48 | 98,14 | 98,55 |
| 3 | 99,72 | 99,53 | 100,00 | 100,00 |
| 4 | 99,63 | 99,59 | 99,67 | 99,59 |
| 5 | 99,34 | 99,55 | 99,14 | 99,11 |
| 6 | 100,00 | 100,00 | 100,00 | 100,00 |
| 7 | 93,88 | 89,33 | 98,61 | 98,53 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 211 | 4 | 7 | 271 |
| 14 | 144 | 0 | 1 | 214 |
| 15 | 304 | 1 | 1 | 241 |
| 16 | 230 | 2 | 1 | 222 |
| 17 | 332 | 0 | 0 | 297 |
| 18 | 213 | 3 | 24 | 201 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 99,19 | 99,28 | 99,07 | 99,28 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 99,82 | 100,00 | 99,67 | 99,59 |
| 5 | 99,78 | 99,55 | 100,00 | 100,00 |
| 6 | 99,21 | 99,33 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 213 | 2 | 2 | 276 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 304 | 1 | 0 | 242 |
| 16 | 232 | 0 | 1 | 222 |
| 17 | 329 | 3 | 2 | 295 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 94,93 | 95,32 | 94,42 | 95,67 |
| 3 | 92,20 | 87,91 | 98,61 | 98,95 |
| 4 | 99,82 | 100,00 | 99,67 | 99,59 |
| 5 | 98,68 | 99,10 | 98,28 | 98,22 |
| 6 | 99,84 | 100,00 | 99,70 | 99,66 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 203 | 12 | 13 | 265 |
| 14 | 142 | 2 | 26 | 189 |
| 15 | 304 | 1 | 0 | 242 |
| 16 | 228 | 4 | 2 | 221 |
| 17 | 331 | 1 | 0 | 297 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 94,73 | 93,88 | 95,81 | 96,67 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,12 | 98,65 | 99,57 | 99,55 |
| 6 | 97,62 | 95,29 | 99,70 | 99,65 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 100,00 | 99,65 | 99,75 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 206 | 9 | 17 | 261 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 231 | 1 | 3 | 220 |
| 17 | 331 | 1 | 14 | 283 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 287 | 1 | 0 | 395 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,96 | 95,32 | 99,07 | 99,25 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 99,82 | 99,59 | 100,00 | 100,00 |
| 5 | 99,78 | 99,55 | 100,00 | 100,00 |
| 6 | 99,52 | 99,33 | 99,70 | 99,66 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 213 | 2 | 13 | 265 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 1 | 241 |
| 16 | 232 | 0 | 1 | 222 |
| 17 | 331 | 1 | 2 | 295 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 97,57 | 96,04 | 99,53 | 99,63 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 99,55 | 99,14 | 99,11 |
| 6 | 99,84 | 100,00 | 99,70 | 99,66 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 99,83 | 100,00 | 99,66 | 99,66 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 214 | 1 | 11 | 267 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 230 | 2 | 1 | 222 |
| 17 | 331 | 1 | 0 | 297 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 296 | 1 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 96,15 | 97,12 | 94,88 | 96,09 |
| 3 | 99,72 | 100,00 | 99,31 | 99,54 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 99,05 | 98,99 | 99,10 | 98,99 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 99,75 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 204 | 11 | 8 | 270 |
| 14 | 143 | 1 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 329 | 3 | 3 | 294 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 1 | 394 |
| 20 | 0 | 297 | 0 | 297 |

A continuación, se muestra las gráficas de las dos primeras iteraciones:

- **Primera iteración:**

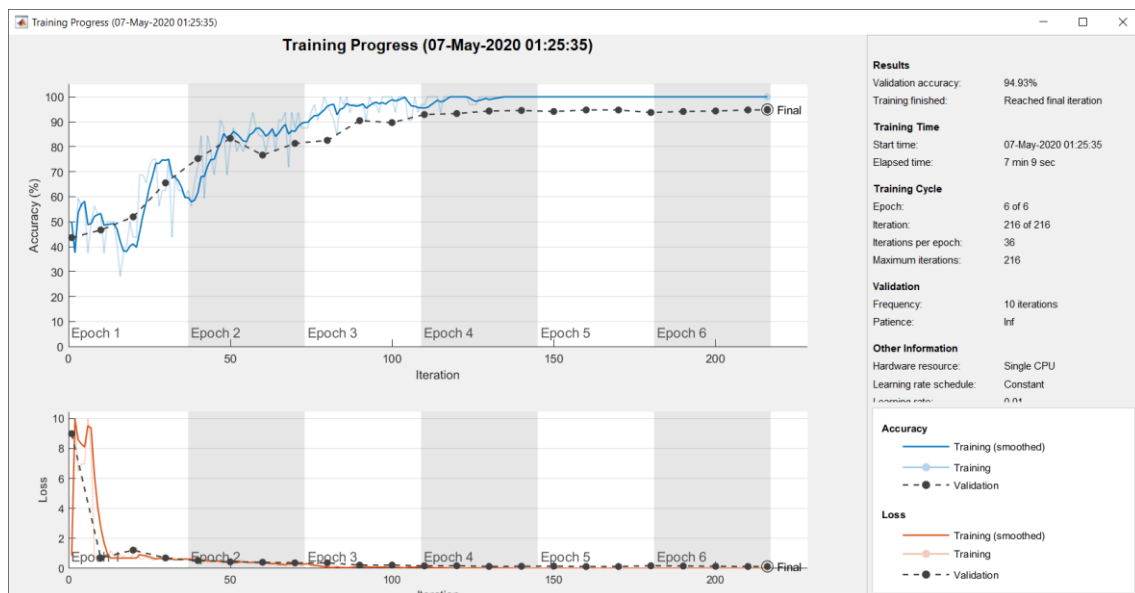


Imagen 197. Cámara Hospital1-Barranoseco.

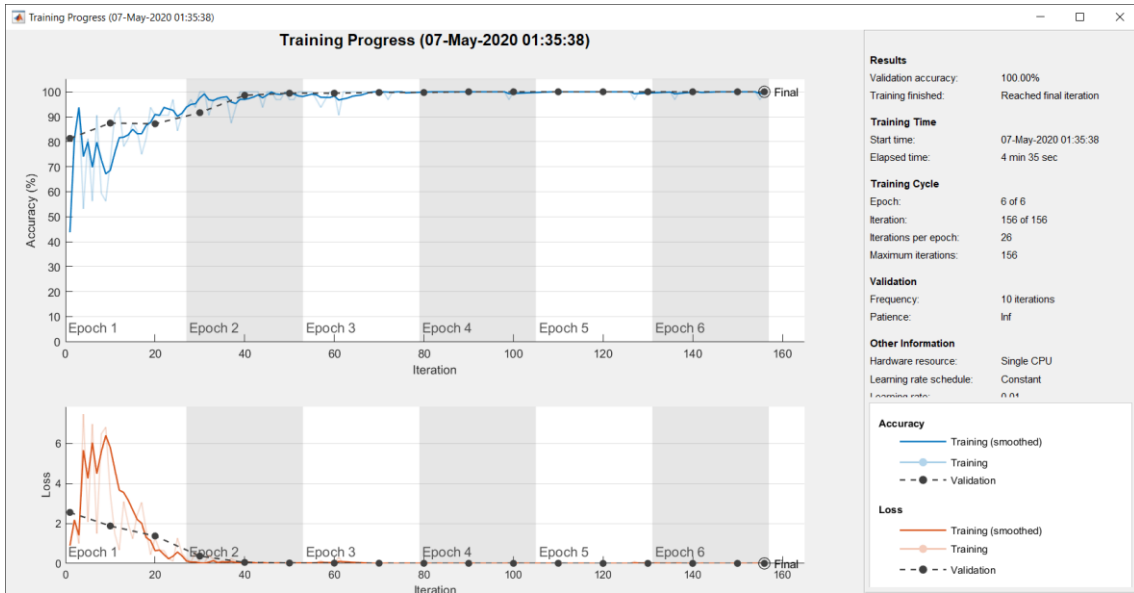


Imagen 198. Cámara Hospital2-Catedral.

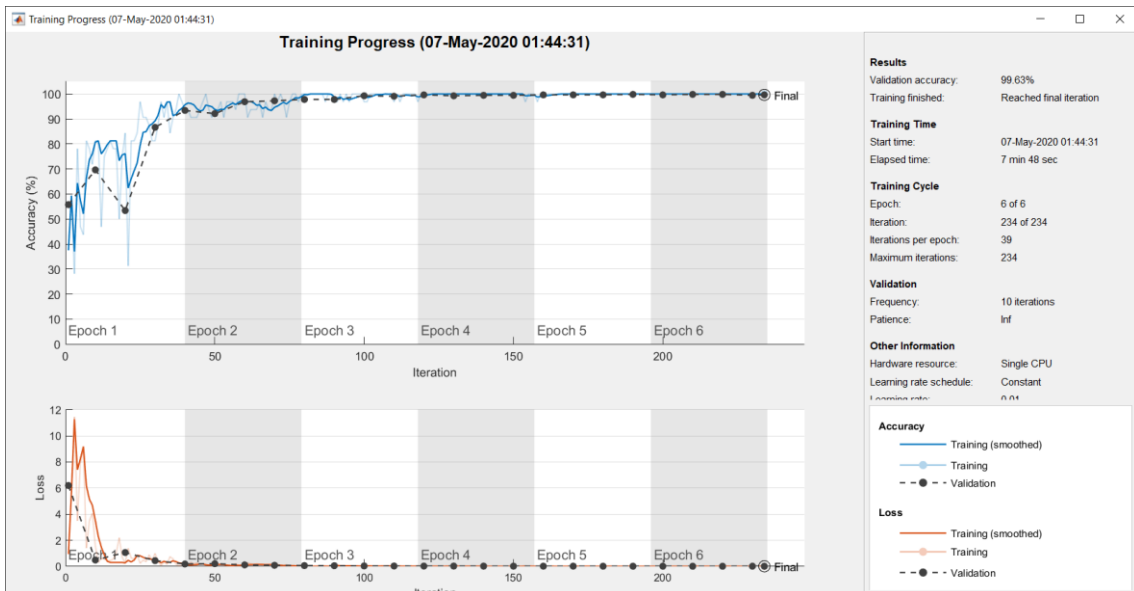


Imagen 199. Cámara Hospital3-Isleta.

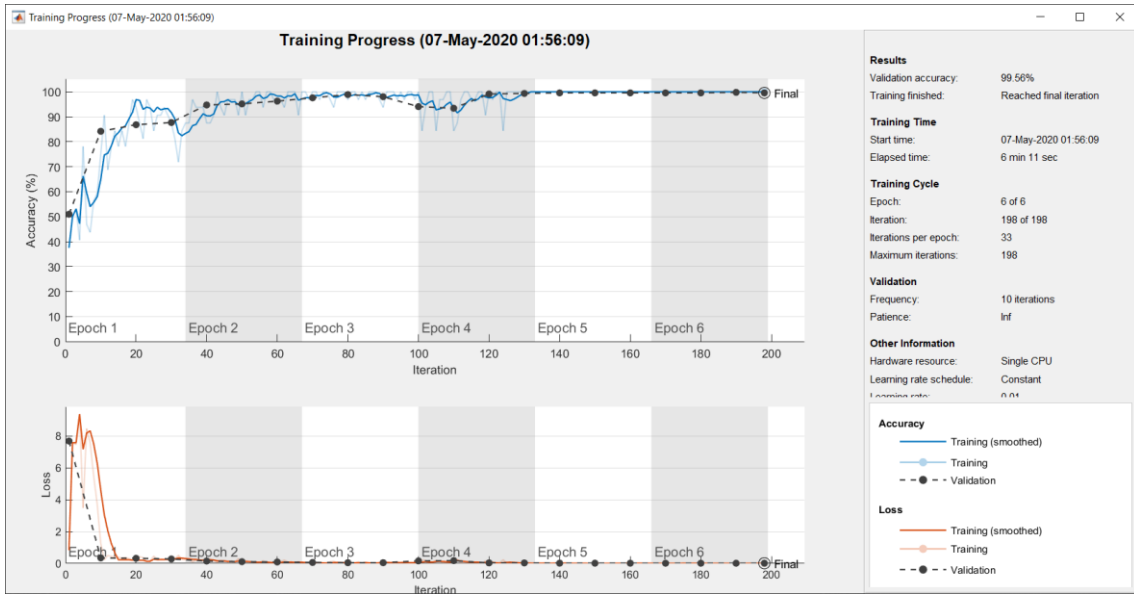


Imagen 200. Cámara Hospital4-Escaleritas.

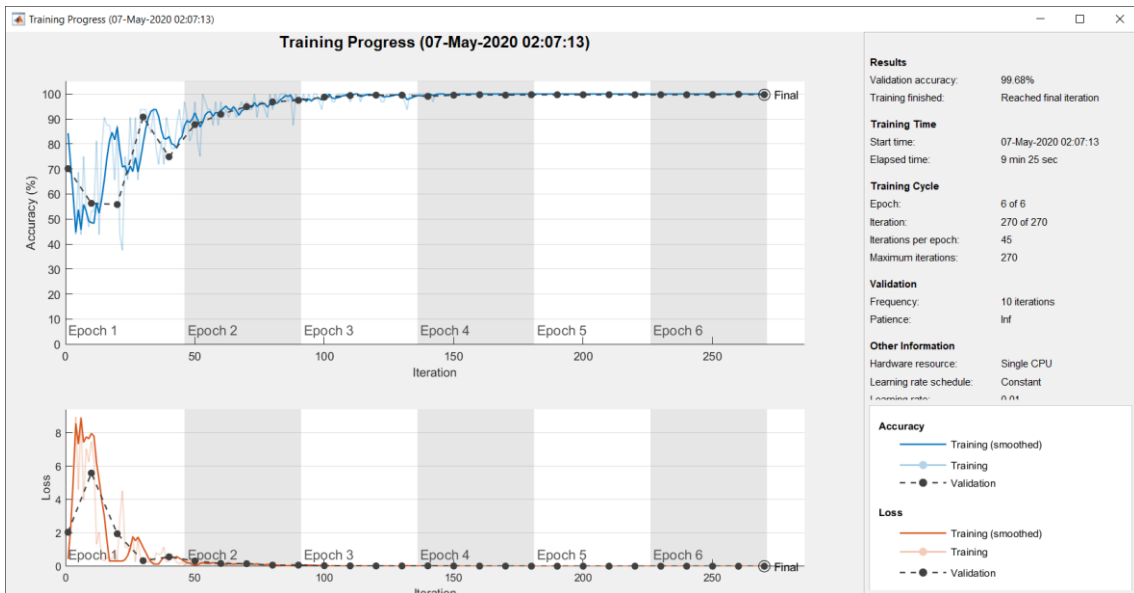


Imagen 201. Cámara Tafira1-Informática.

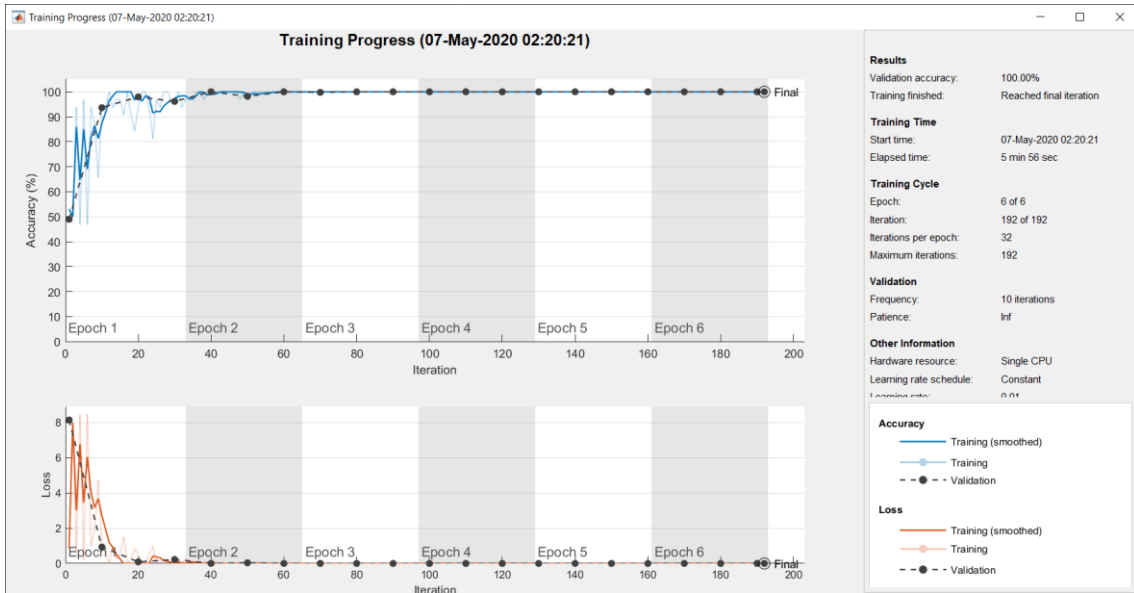


Imagen 202. Cámara Tafira2-MontañaArucas.

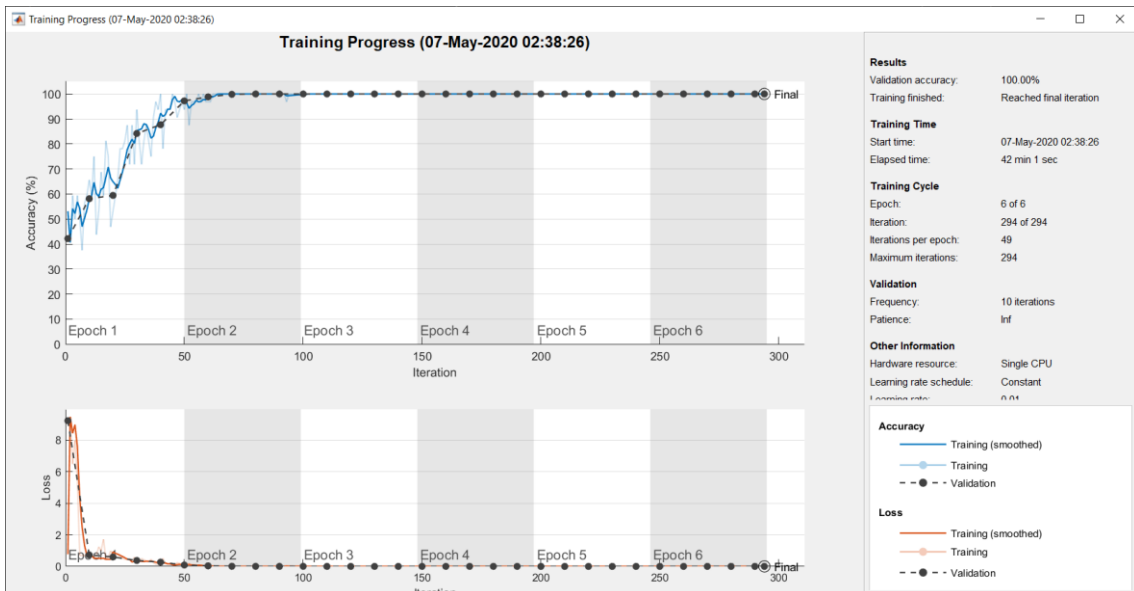


Imagen 203. Cámara Tafira3-Foscam.

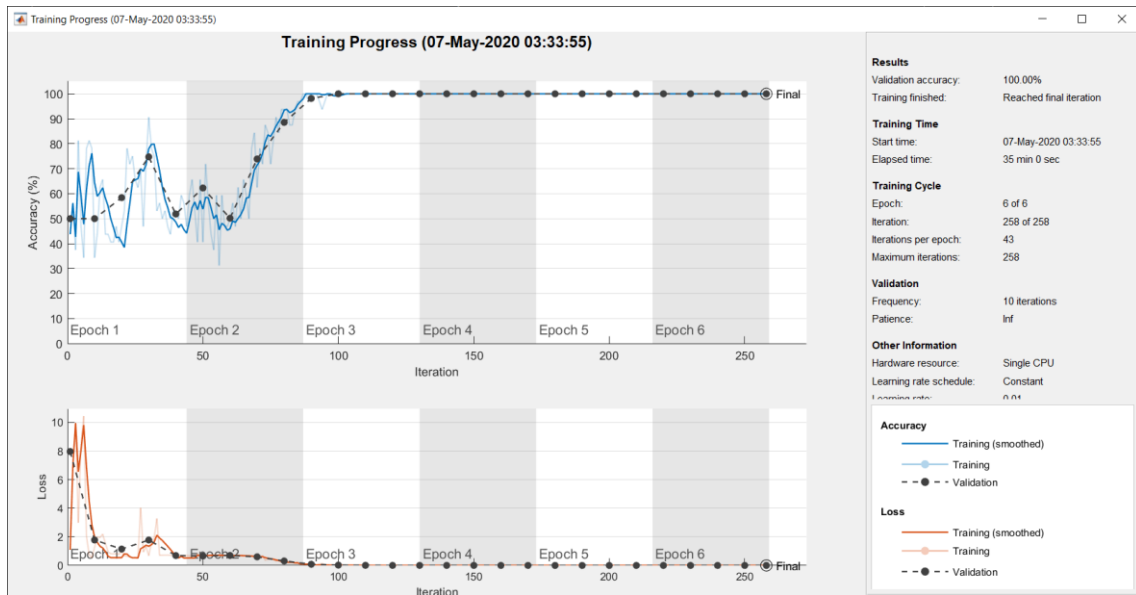


Imagen 204. Cámara Tafira4-Foscam.

- Segunda iteración:

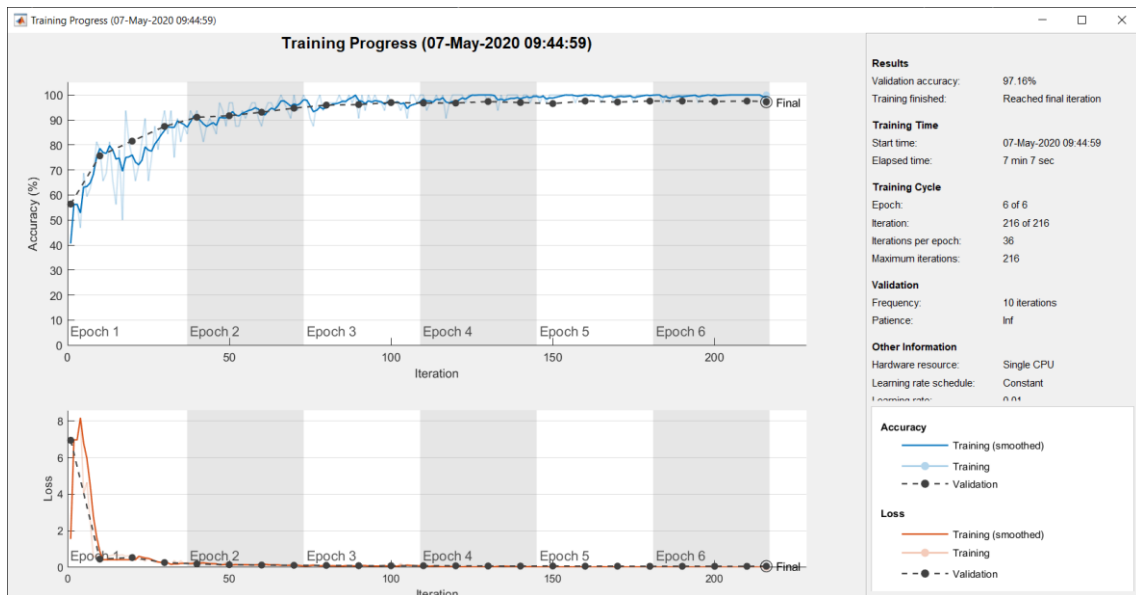


Imagen 205. Cámara Hospital1-Barranoseco.

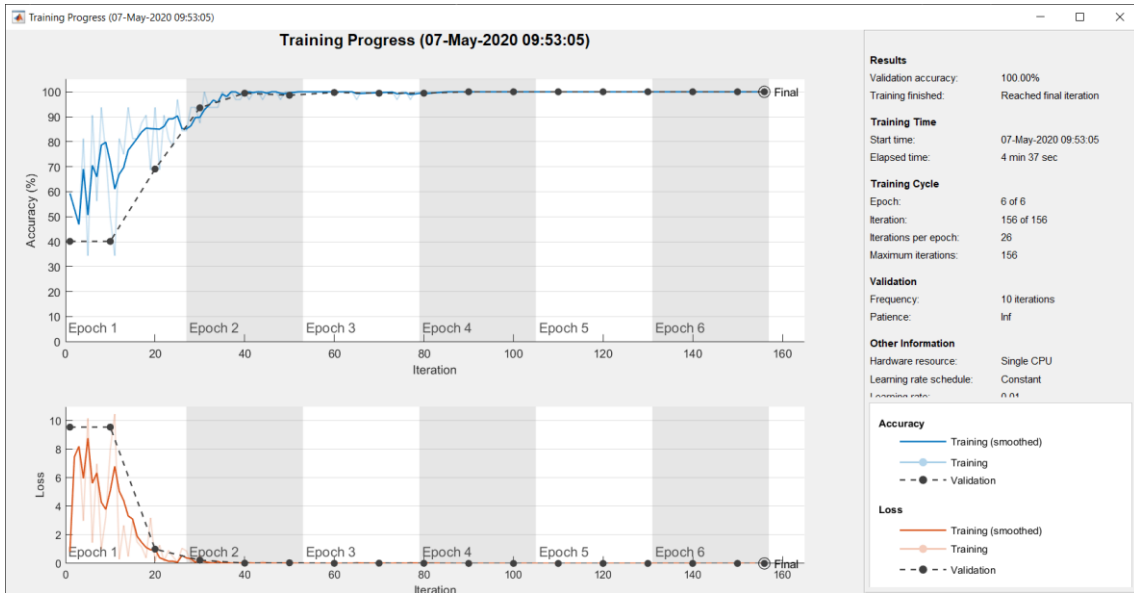


Imagen 206. Cámara Hospital2-Catedral.

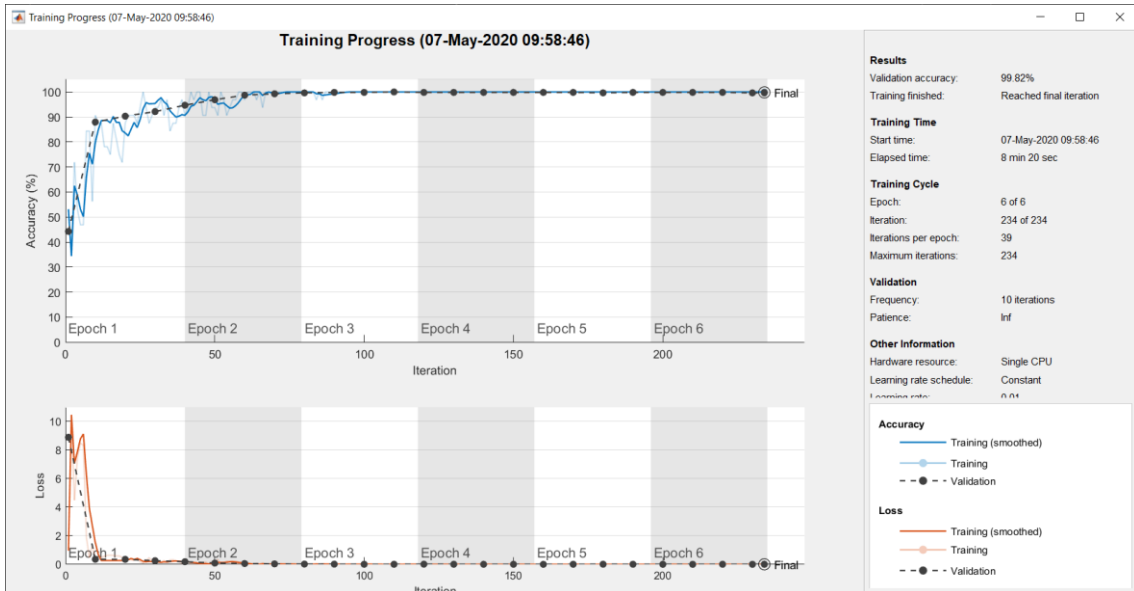


Imagen 207. Cámara Hospital3-Isleta.

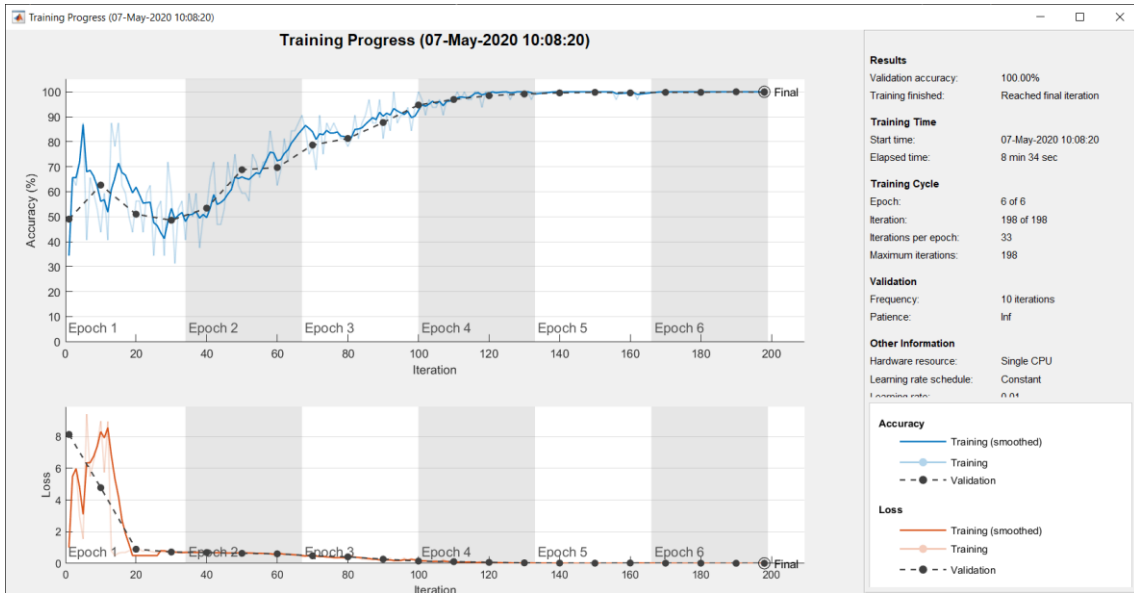


Imagen 208. Cámara Hospital4-Escaleritas.

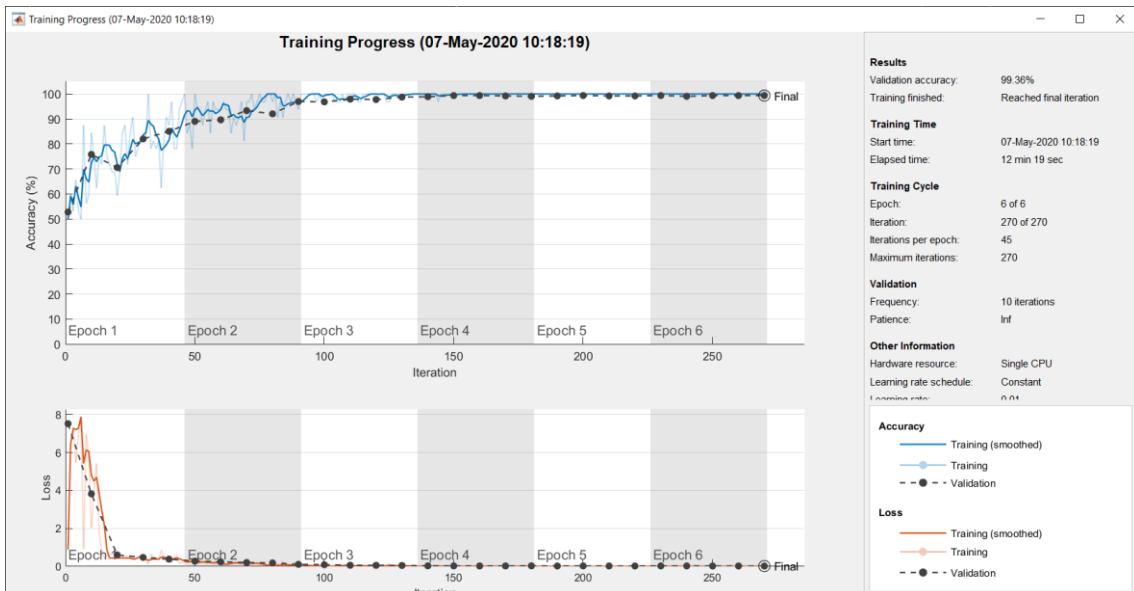


Imagen 209. Cámara Tafira1-Informática.

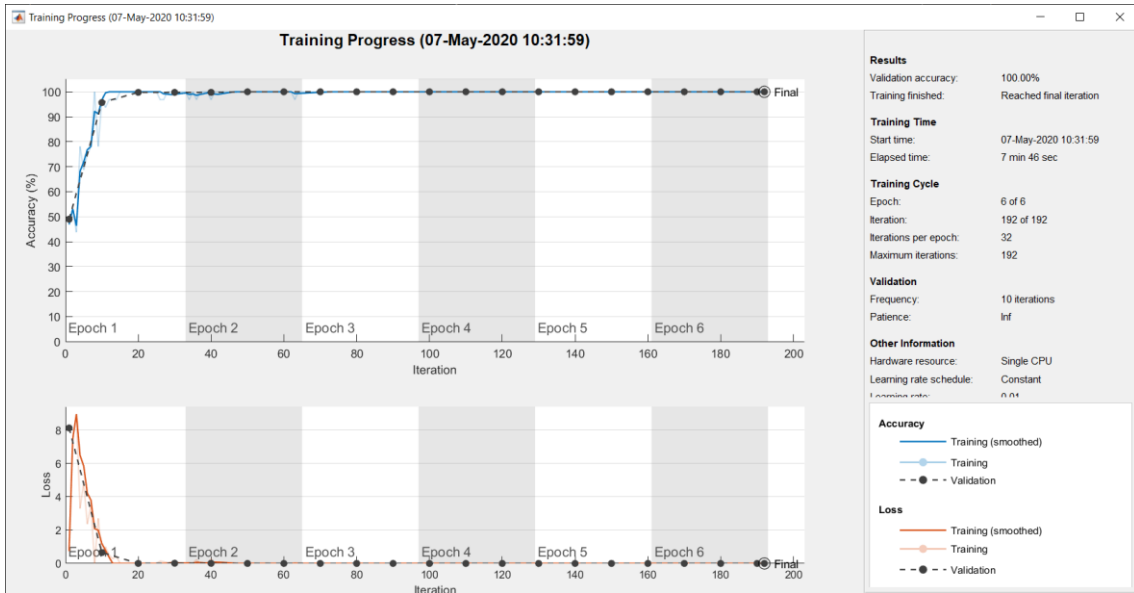


Imagen 210. Cámara Tafira2-MontañaArucas.

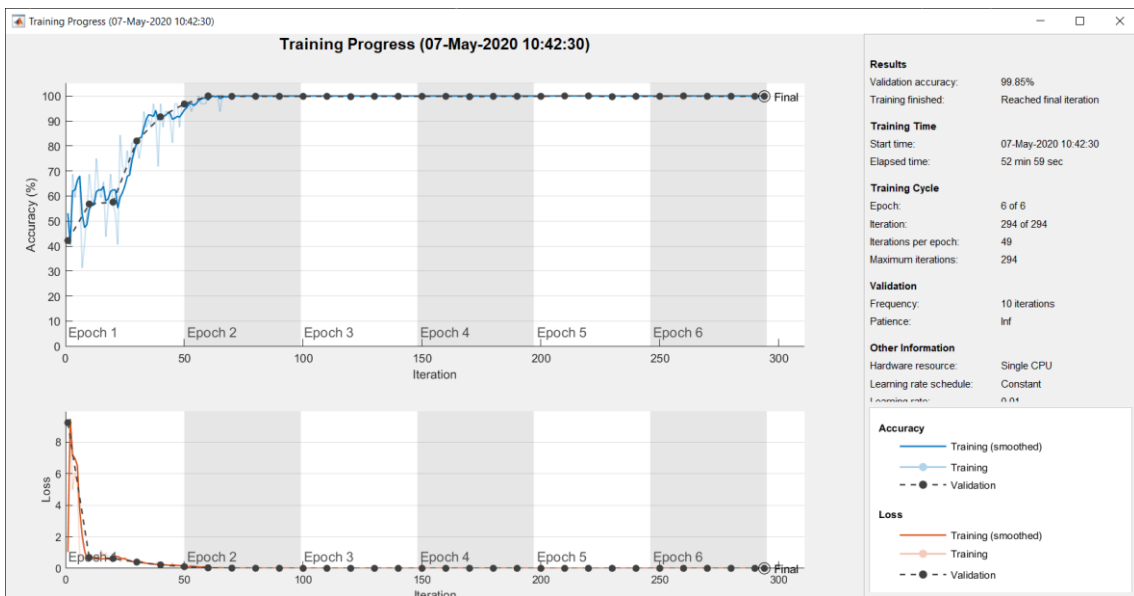


Imagen 211. Cámara Tafira3-Foscam.

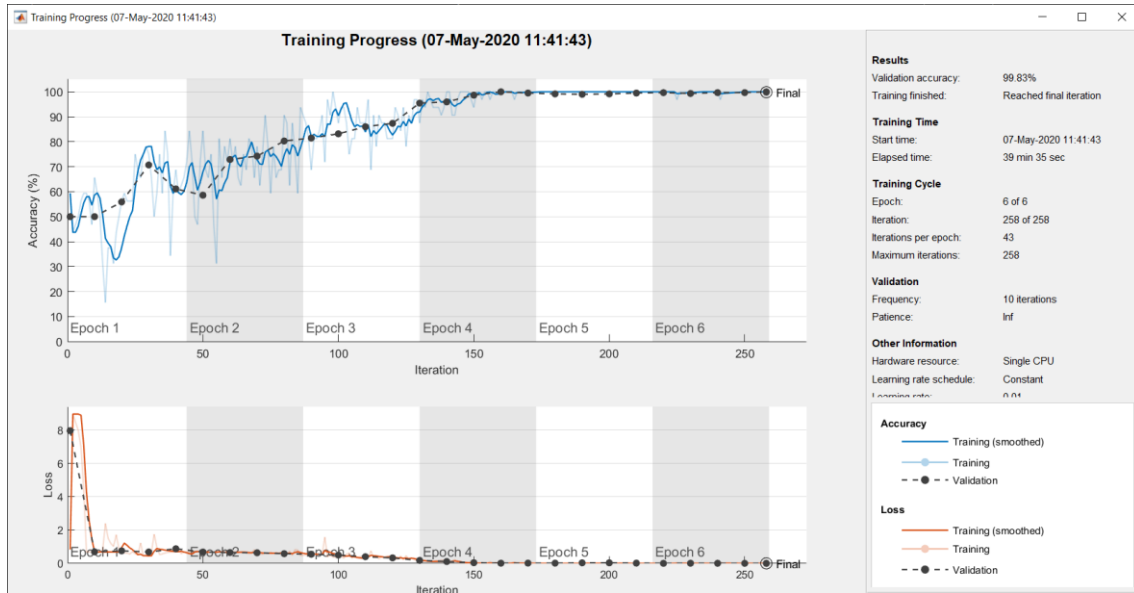


Imagen 212. Cámara Tafira4-Foscam.

- Resultados usando el modo de color Cielab y se realizaron 9 iteraciones. En la Tabla 19 se muestra 9 iteraciones.

Tabla 19. Resultados de las 9 iteraciones con el modo de color Cielab.

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 56,39 | 100,00 | 0,00 | 56,39 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 97,14 | 94,17 | 100,00 | 100,00 |
| 6 | 79,65 | 56,90 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 93,94 | 87,88 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 0 | 215 | 0 | 278 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 232 | 0 | 13 | 210 |
| 17 | 332 | 0 | 128 | 169 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 297 | 0 | 36 | 261 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 89,05 | 93,88 | 82,79 | 87,58 |
| 3 | 96,94 | 100,00 | 92,36 | 95,13 |
| 4 | 99,45 | 98,76 | 100,00 | 100,00 |
| 5 | 96,70 | 93,27 | 100,00 | 100,00 |
| 6 | 81,72 | 61,62 | 99,70 | 99,46 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 93,70 | 100,00 | 85,07 | 90,18 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 178 | 37 | 17 | 261 |
| 14 | 133 | 11 | 0 | 215 |
| 15 | 305 | 0 | 3 | 239 |
| 16 | 232 | 0 | 15 | 208 |
| 17 | 331 | 1 | 114 | 183 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 245 | 43 | 0 | 395 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 64,50 | 97,84 | 21,40 | 61,68 |
| 3 | 99,44 | 100,00 | 98,61 | 99,08 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,34 | 100,00 | 98,71 | 98,67 |
| 6 | 82,35 | 62,63 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,71 | 99,49 | 100,00 | 100,00 |
| 9 | 79,63 | 89,23 | 70,03 | 74,86 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 46 | 169 | 6 | 272 |
| 14 | 142 | 2 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 0 | 223 |
| 17 | 332 | 0 | 111 | 186 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 2 | 393 |
| 20 | 208 | 89 | 32 | 265 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 56,39 | 100,00 | 0,00 | 56,39 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,56 | 100,00 | 99,14 | 99,11 |
| 6 | 86,96 | 72,73 | 99,70 | 99,54 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 88,43 | 100,00 | 72,57 | 83,33 |
| 9 | 49,49 | 0,00 | 98,99 | 0,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 0 | 215 | 0 | 278 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 230 | 2 | 0 | 223 |
| 17 | 331 | 1 | 81 | 216 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 209 | 79 | 0 | 395 |
| 20 | 294 | 3 | 297 | 0 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 91,28 | 91,37 | 91,16 | 93,04 |
| 3 | 98,33 | 100,00 | 95,83 | 97,29 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 99,78 | 99,55 | 100,00 | 100,00 |
| 6 | 79,97 | 57,58 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 98,98 | 98,73 | 99,31 | 99,49 |
| 9 | 99,83 | 99,66 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 196 | 19 | 24 | 254 |
| 14 | 138 | 6 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 232 | 0 | 1 | 222 |
| 17 | 332 | 0 | 126 | 171 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 286 | 2 | 5 | 390 |
| 20 | 297 | 0 | 1 | 296 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 56,39 | 100,00 | 0,00 | 56,39 |
| 3 | 97,21 | 100,00 | 93,06 | 95,56 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 88,79 | 85,20 | 92,24 | 91,35 |
| 6 | 76,31 | 50,84 | 99,10 | 98,05 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 57,83 | 100,00 | 0,00 | 57,83 |
| 9 | 99,16 | 98,65 | 99,66 | 99,66 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 0 | 215 | 0 | 278 |
| 14 | 134 | 10 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 214 | 18 | 33 | 190 |
| 17 | 329 | 3 | 146 | 151 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 0 | 288 | 0 | 395 |
| 20 | 296 | 1 | 4 | 293 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 86,21 | 94,24 | 75,81 | 83,44 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 98,24 | 97,76 | 98,71 | 98,64 |
| 6 | 79,49 | 56,57 | 100,00 | 100,00 |
| 7 | 99,32 | 99,11 | 99,54 | 99,55 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 50,00 | 100,00 | 0,00 | 50,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 163 | 52 | 16 | 262 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 229 | 3 | 5 | 218 |
| 17 | 332 | 0 | 129 | 168 |
| 18 | 215 | 1 | 2 | 223 |
| 19 | 288 | 0 | 0 | 395 |
| 20 | 0 | 297 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 97,97 | 96,40 | 100,00 | 100,00 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 |
| 5 | 98,68 | 97,31 | 100,00 | 100,00 |
| 6 | 90,46 | 80,13 | 99,70 | 99,58 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 99,85 | 100,00 | 99,65 | 99,75 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 215 | 0 | 10 | 268 |
| 14 | 144 | 0 | 0 | 215 |
| 15 | 305 | 0 | 0 | 242 |
| 16 | 232 | 0 | 6 | 217 |
| 17 | 331 | 1 | 59 | 238 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 287 | 1 | 0 | 395 |
| 20 | 297 | 0 | 0 | 297 |

| | A | B | C | D |
|----|----------------------|-----------------------------|-----------------------------|-----------------------|
| 1 | Accuracy (Exactitud) | Sensibilidad (Sensibilidad) | Specificity (Especificidad) | Precision (Precisión) |
| 2 | 62,47 | 100,00 | 13,95 | 60,04 |
| 3 | 98,33 | 100,00 | 95,83 | 97,29 |
| 4 | 55,76 | 0,00 | 100,00 | |
| 5 | 98,02 | 96,41 | 99,57 | 99,54 |
| 6 | 81,72 | 61,62 | 99,70 | 99,46 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 98,54 | 97,47 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 |
| 10 | | | | |
| 11 | | | | |
| 12 | TN | FP | FN | TP |
| 13 | 30 | 185 | 0 | 278 |
| 14 | 138 | 6 | 0 | 215 |
| 15 | 305 | 0 | 242 | 0 |
| 16 | 231 | 1 | 8 | 215 |
| 17 | 331 | 1 | 114 | 183 |
| 18 | 216 | 0 | 0 | 225 |
| 19 | 288 | 0 | 10 | 385 |
| 20 | 297 | 0 | 0 | 297 |

A continuación, se muestra las gráficas de las dos primeras iteraciones:

- **Primera iteración:**

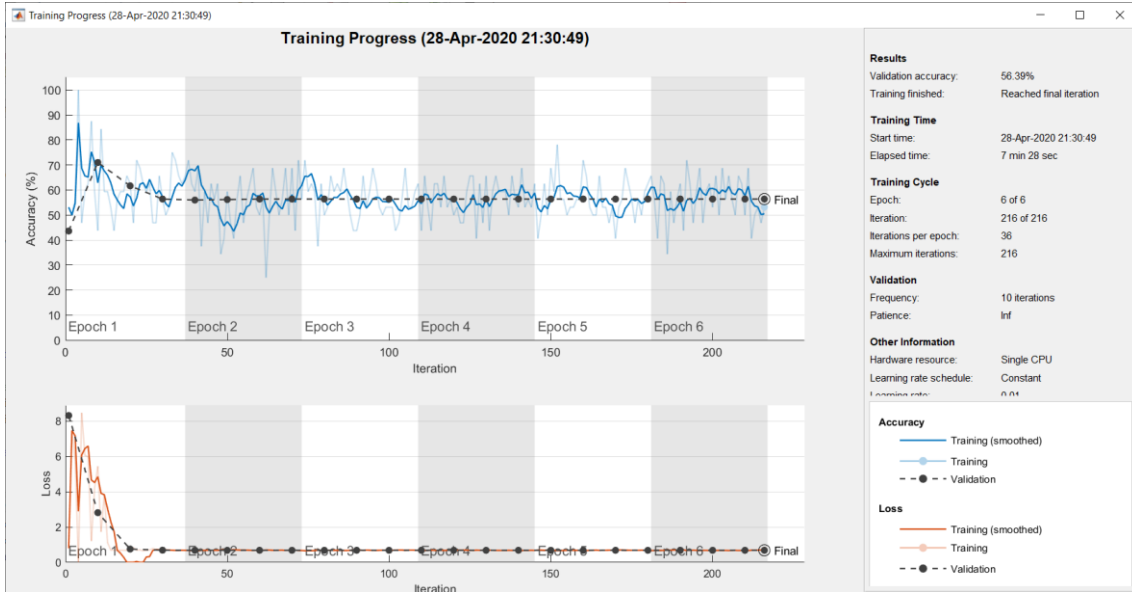


Imagen 213. Cámara Hospital1-Barranoseco.

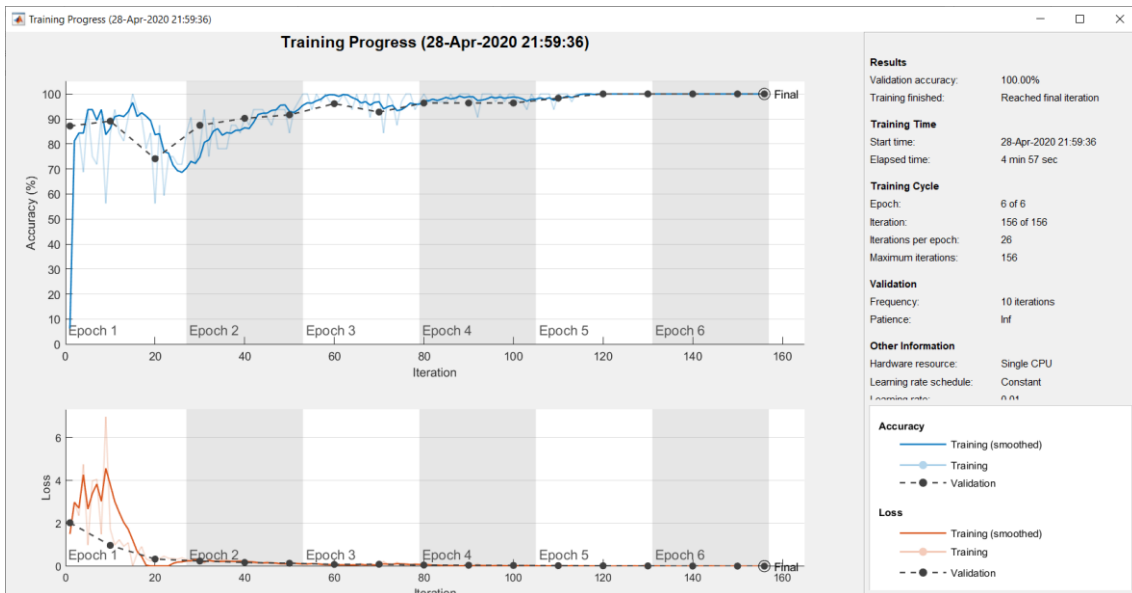


Imagen 214. Cámara Hospital2-Catedral.

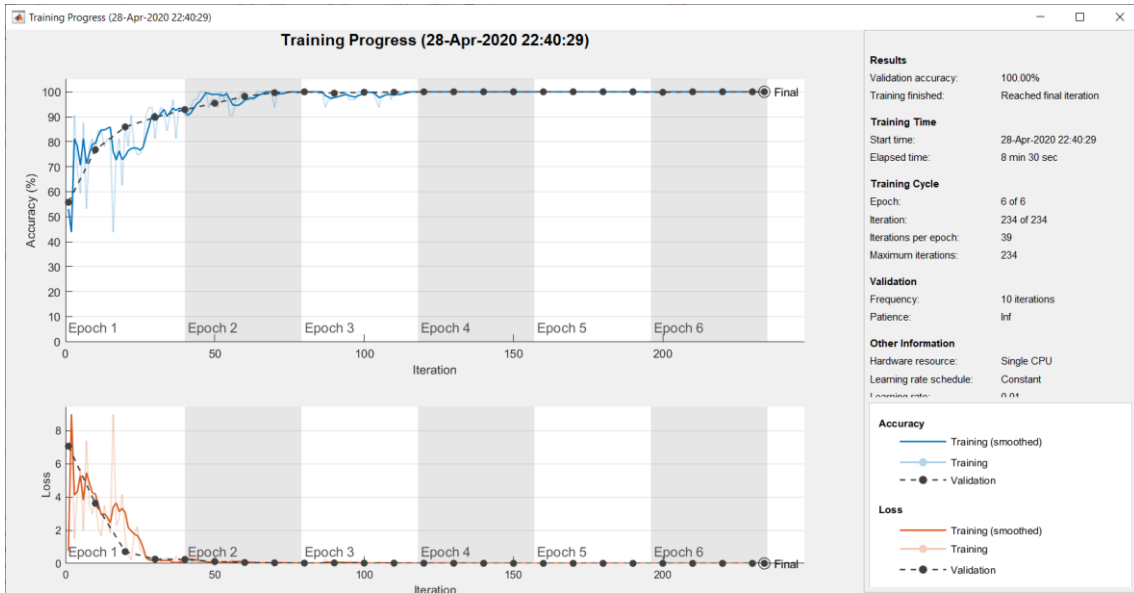


Imagen 215. Cámara Hospital3-Isleta.

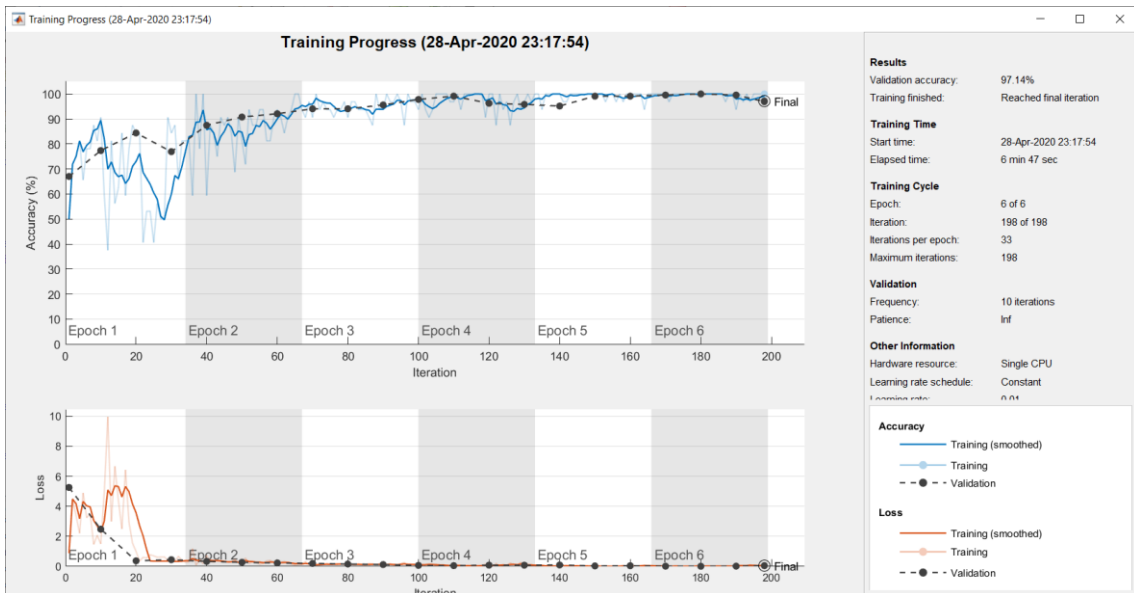


Imagen 216. Cámara Hospital4-Escaleritas.

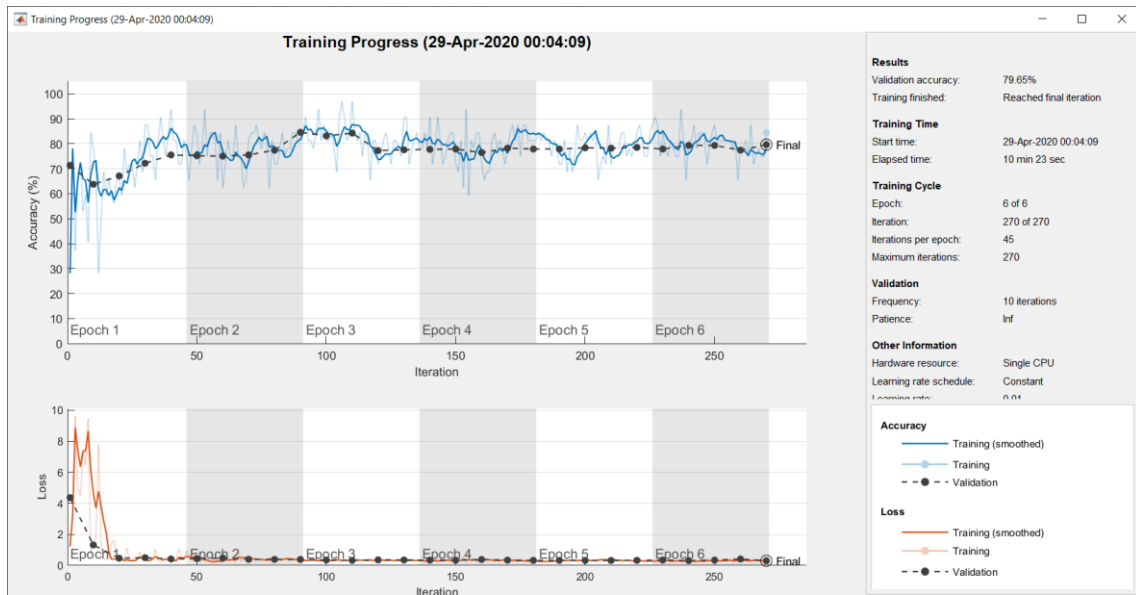


Imagen 217. Cámara Tafira1-Infomática.

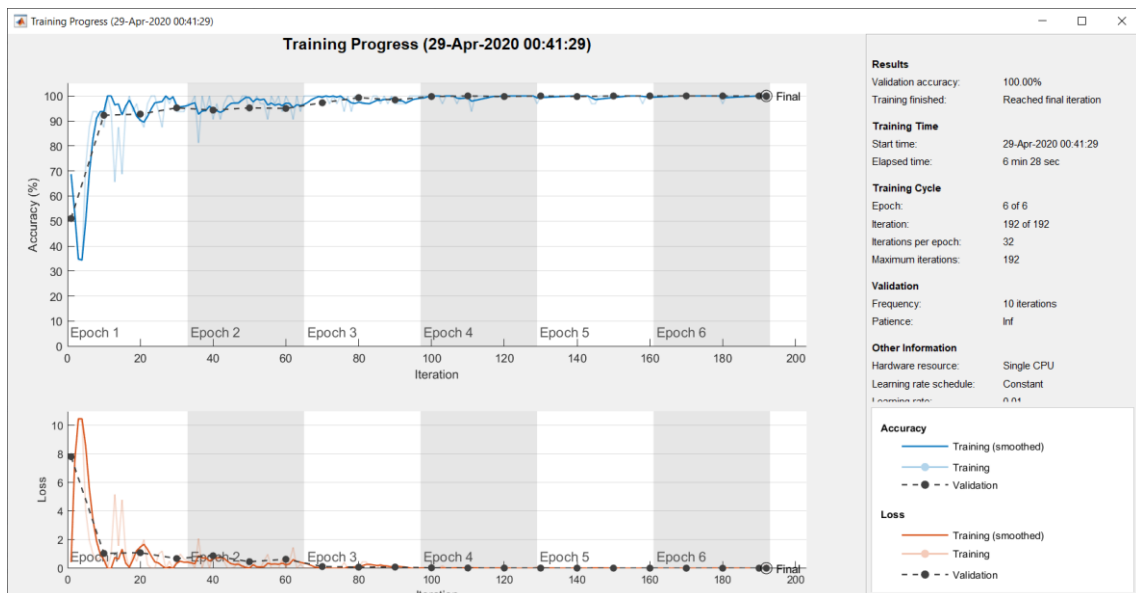


Imagen 218. Cámara Tafira2-MontañaArucas.

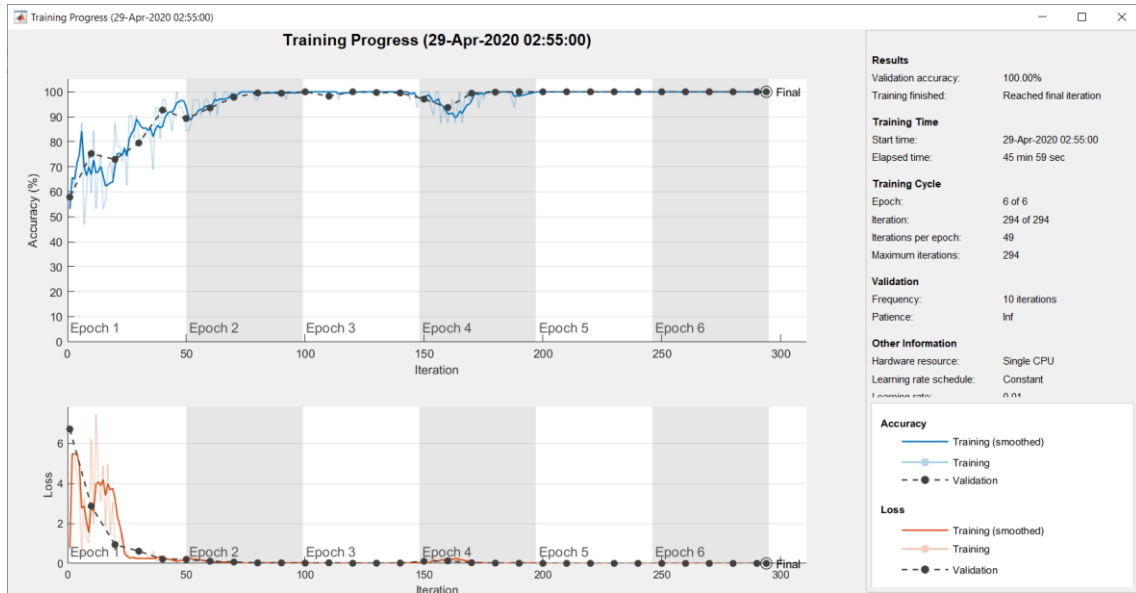


Imagen 219. Cámara Tafira3-Foscam.

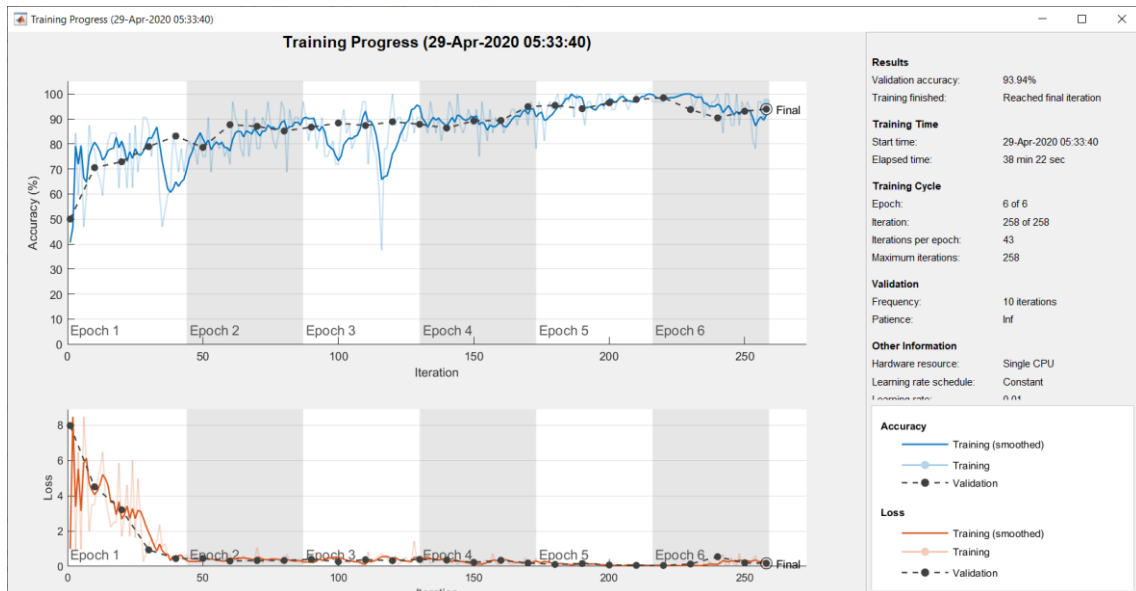


Imagen 220. Cámara Tafira4-Foscam.

- Segunda iteración:

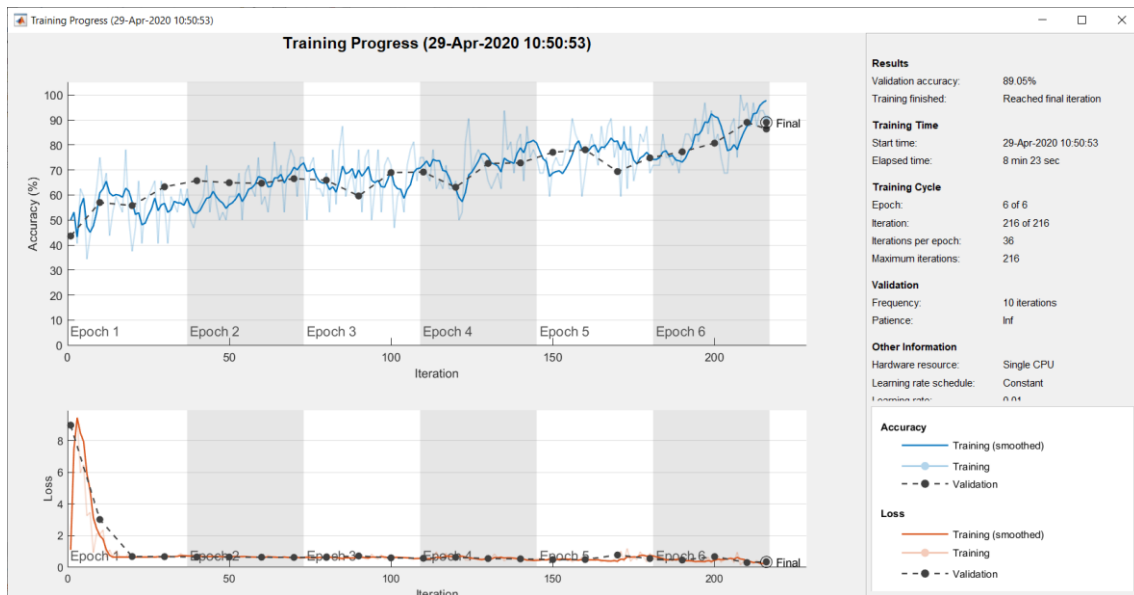


Imagen 221. Cámara Hospital1-Barranoseco.

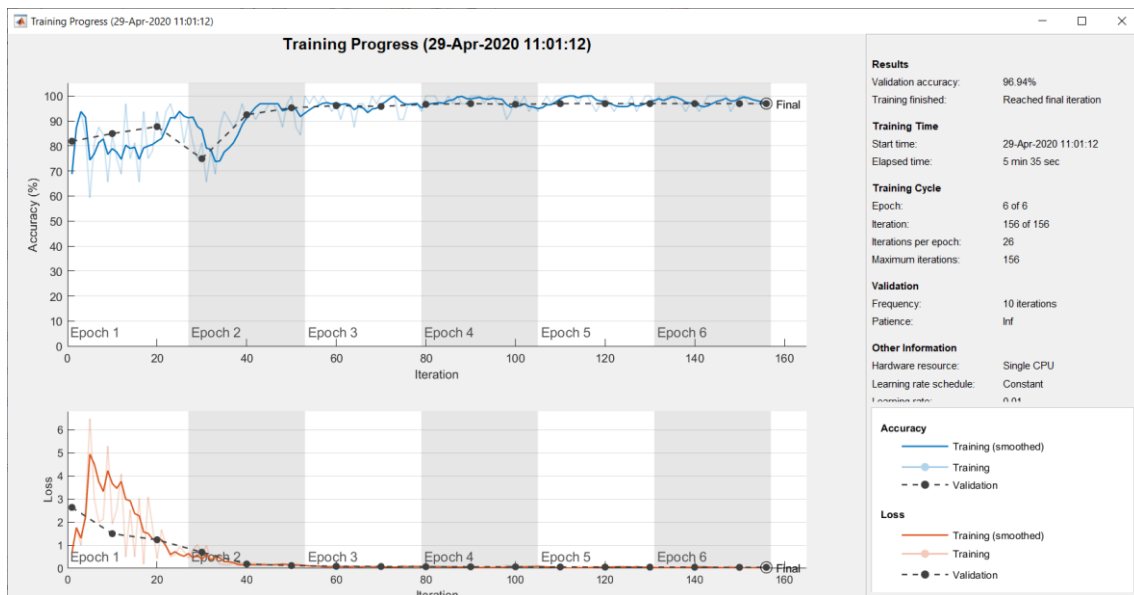


Imagen 222. Cámara Hospital2-Catedral.

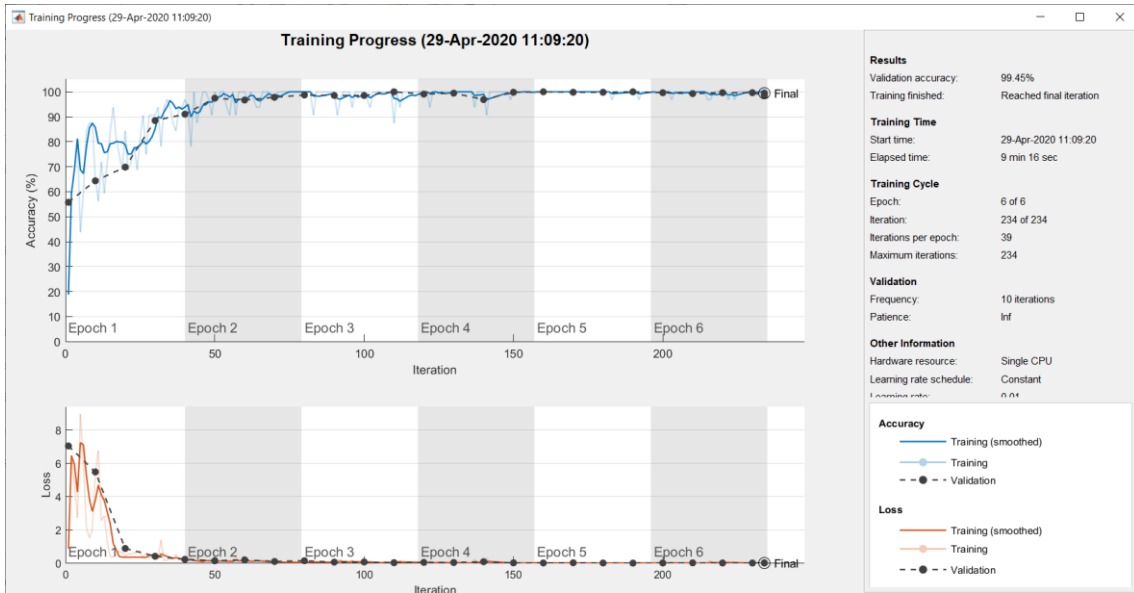


Imagen 223. Cámara Hospital3-Isleta.

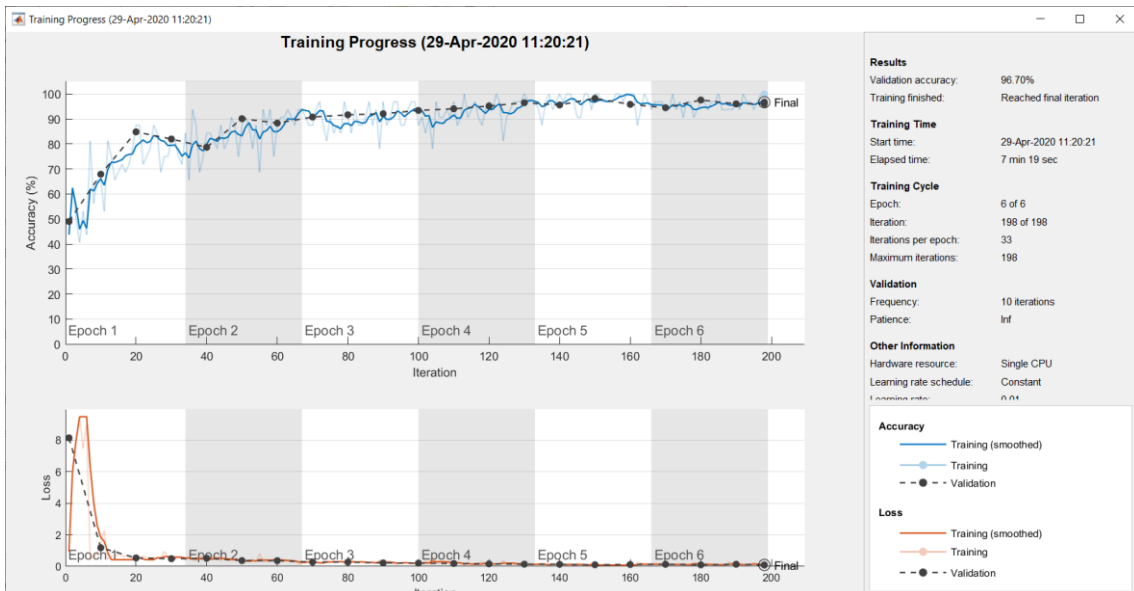


Imagen 224. Cámara Hospital4-Escaleritas.

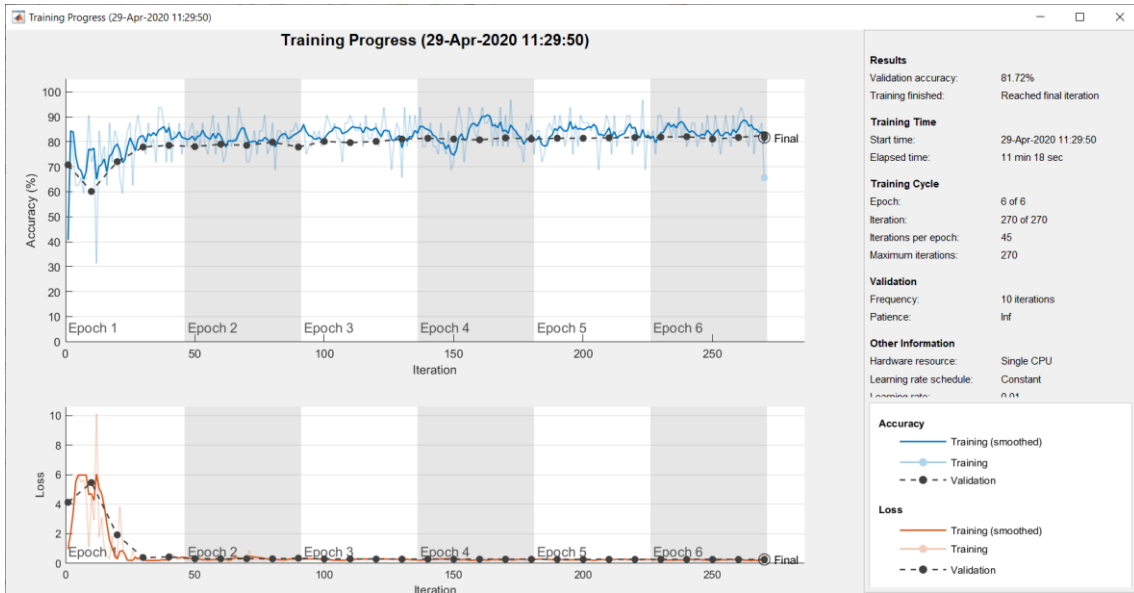


Imagen 225. Cámara Tafira1-Infomática.

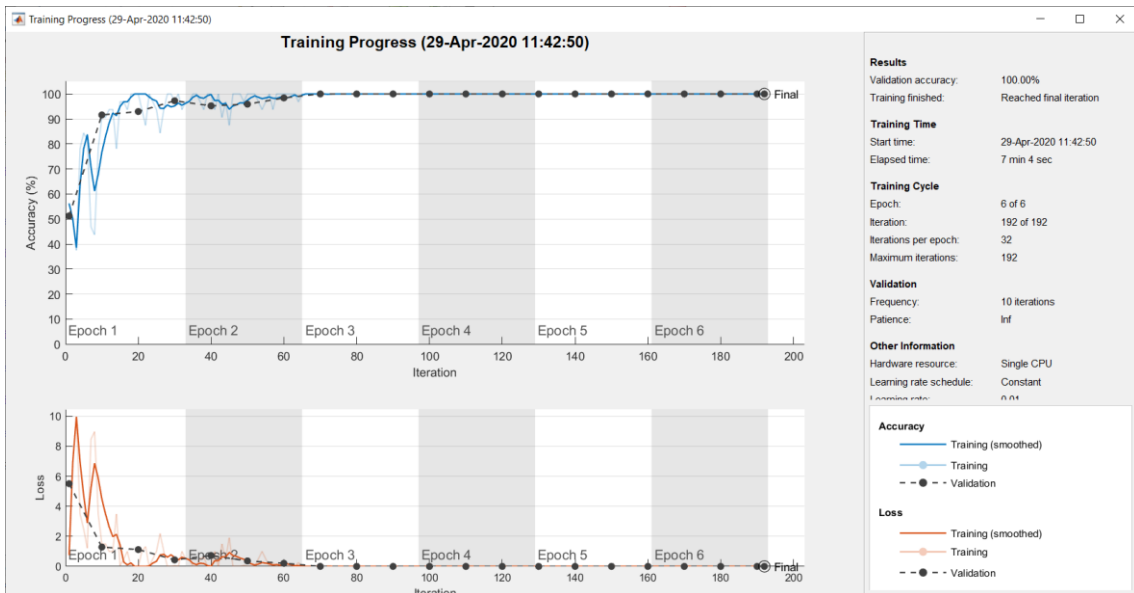


Imagen 226. Cámara Tafira2-MontañaArucas.

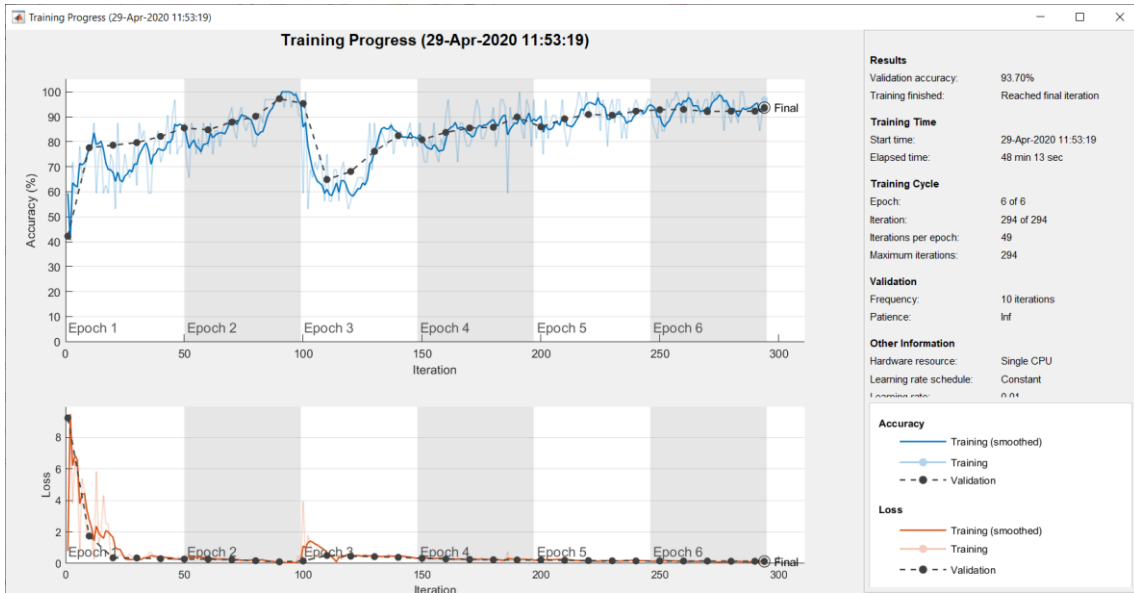


Imagen 227. Cámara Tafira3-Foscam.

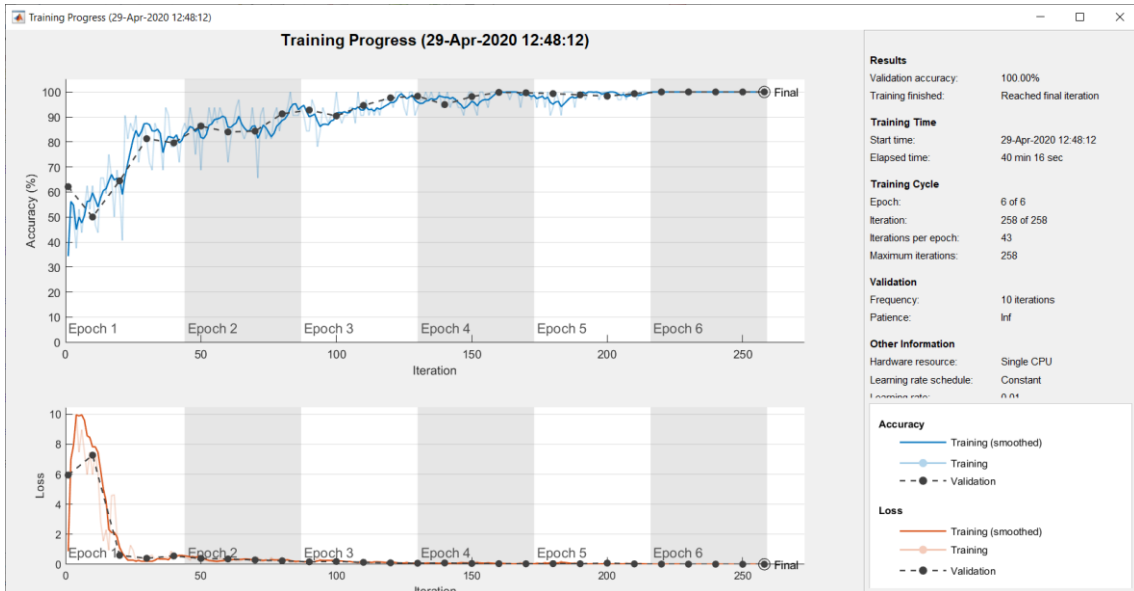


Imagen 228. Cámara Tafira4-Foscam.

Bibliografía

[AEMET 2021] Página oficial de la Agencia Estatal de meteorología de España. Disponible en:

<http://www.aemet.es/es/portada>

Fecha de revisión junio del 2021.

[AEMET 2020] Agencia Estatal de meteorología de España. Disponible en:

https://meteoglosario.aemet.es/es/termino/990_ciclo-hidrologico

Fecha de revisión junio del 2021.

[AEMET 2018] Agencia Estatal de meteorología de España. Disponible en:

https://meteoglosario.aemet.es/es/termino/313_lluvia-engelante

https://meteoglosario.aemet.es/es/termino/243_lluvia

Fecha de revisión septiembre del 2020.

[AEMET 2009] Agencia Estatal de meteorología de España. Disponible en:

http://izana.aemet.es/index.php?option=com_content&view=article&id=208%3A-development-of-a-pluviometer-to-measure-water-content-of-mist&catid=10%3Anews&Itemid=49&lang=es

Fecha de revisión septiembre del 2020.

[Aguilar 2014] Aguilar, Wilbert; Angulo Bahón, Cecilio. Estabilización de vídeo en micro vehículos aéreos y su aplicación en la detección de caras. En Revista Digital Congreso de Ciencia y Tecnología: Memorias. Sesiones Técnicas. 2014. p. 155-160.

[Bronte 2009] Bronte, S., Bergasa, L.M., Alcantarilla, P.F.: "Fog Detection System Based on Computer Vision Techniques". In: 12th International IEEE Conference on Intelligent Transportation Systems (2009).

[Brussaard 1995] G. Brussaard, P. A. Watson and P. A. Watson, Atmospheric Modelling and Millimetre Wave Propagation. 1995.

[Díaz 2021] Darío Díaz Caballero. Trabajo Fin de Grado "Técnicas de procesamiento de imágenes de videovigilancia aplicadas a la detección de lluvia mediante aprendizaje máquina". Tutor: Dr. D. Juan Luis Navarro Mesa. Grado en Ingeniería en Tecnologías de la Telecomunicación. Escuela de Ingeniería de Telecomunicación y Electrónica. Universidad de Las Palmas de Gran Canaria. Enero 2021.

[Duda 1973] Duda R. and Hart P. "Pattern Classification and Scene Analysis". John Wiley and Sons, 1973.

[Fuente 2020] Sonia Fuente González. Trabajo Fin de Grado "Técnicas de aprendizaje máquina para detección y cuantificación de lluvia mediante radioenlaces terrenos a frecuencia de gigahercios". Tutores: Dr. D. Juan Luis Navarro Mesa y Dr. D. Eduardo Hernández Pérez. Grado en Ingeniería en Tecnologías de la Telecomunicación. Escuela de Ingeniería de

Telecomunicación y Electrónica. Universidad de Las Palmas de Gran Canaria. Julio de 2020.

[Godoy 2019] Godoy Rosario, José A., Quintana Morales, Pedro J., Ravelo García, Antonio G. y Navarro Mesa, Juan L. "An Approach to Rain Detection using Sobel Image Preprocessing and Convolutional Neuronal Networks". 2019. Universidad de Las Palmas de Gran Canaria.

[González 2017] Patricia del Carmen González Gómez. Proyecto Fin de Carrera "Diseño de técnicas de estimación de lluvia a partir de niveles de potencia de radioenlaces". Tutores: Dr. D. Juan Luis Navarro Mesa y Dr. D. Iván D. Guerra Moreno. Ingeniería de Telecomunicación. Escuela de Ingeniería de Telecomunicación y Electrónica. Universidad de Las Palmas de Gran Canaria. Año 2017.

[Goodfellow 2016] Ian Goodfellow, Yoshua Bengio y Aaron Courville, "Deep Learning" MIT Press. 2016.

[Gunn 1949] R. Gunn and G. D. Kinzer, "The terminal velocity of fall for water droplets in stagnant air," J. Meteorol., vol. 6, (4), pp. 243-248, 1949.

[Guerra 2017] Iván D. Guerra Moreno, "Algoritmos De Aprendizaje Aplicados a La Monitorización Meteorológica Mediante Redes Inalámbricas De Comunicaciones Distribuidas.", Tesis Doctoral. Programa de Doctorado en Cibernética y Telecomunicación.

[Hastie 2009] Trevor Hastie, Robert Tibshirani y Jerome Friedman, "The Elements of Statistical Learning" Springer Series in Statistics. 01-10-2009.

[Julio Pardo 2020] Ingeniero Carlos Julio Pardo. Web sobre "La Visión Por Computador". Histograma de gradientes Orientados. Año 2020. Disponible en: <https://carlosjuliopardoblog.wordpress.com/2017/12/04/hog-histograma-de-gradientes-orientados/>
Fecha de revisión junio del 2021.

[Kriesel 2005] David Kriesel, "A brief introduction to Neural Networks". 2005. http://www.dkriesel.com/en/science/neural_networks
Fecha de revisión octubre del 2020.

[Kubat 2017] Miroslav Kubat, "An Introduction to Machine Learning", Springer International Publishing, Second Edition, 2017.

[Manso 2020] Alba Manso Fernández. Trabajo de Fin de Máster "Algoritmo para la identificación temprana de la sepsis a partir de datos clínicos". Tutores: Encarnación Gimeno Nieves y Juan Luis Navarro Mesa. Máster Universitario en Ingeniería de Telecomunicación. Escuela Politécnica Superior. Universidad de Alicante. Septiembre de 2020.

[Manu 2015] Manu, B. N. Rain removal from still images using L0 gradient minimization technique. En 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, 2015. p. 263-268.

[OMM 2019] Publicaciones de la OMM (407). OMM. 1993. p. 3-5. ISBN 92-63-30407-6. Fecha de revisión el 24 de junio del 2021.

[Pruppacher 1971] H. R. Pruppacher y R. Pitter, "A semi-empirical determination of the shape of cloud and rain drops," J. Atmos. Sci., vol. 28, (1), pp. 86-94, 1971.

[Schwendeman 2015] Schwendeman, Michael; Thomson, Jim. A horizon-tracking method for shipboard video stabilization and rectification. Journal of Atmospheric and Oceanic Technology, 2015, vol. 32, no 1, p. 164-176.

[Sebe 2005] Sebe, Nicu, et al. Machine learning in computer vision. Springer Science & Business Media, 2005.

[Skow 2006] Skow, Michael. Automatic white balance for digital imaging. U.S. Patent No 6,995,791, 7 Feb. 2006.

[Sobel 1973] Sobel, Irwin & Feldman, Gary. (1973). "A 3×3 isotropic gradient operator for image processing". Pattern Classification and Scene Analysis. 271-272.

[Sociedad Geográfica de Lima 2011] Sociedad Geográfica de Lima. "Contribuyendo al desarrollo de una Cultura del Agua y la Gestión Integral de Recurso Hídrico" Lima, Perú 2011. Disponible en: https://www.gwp.org/globalassets/global/gwp-sam_files/publicaciones/varios/ciclo_hidrologico.pdf
Fecha de revisión junio del 2021.

[Soo-Kyun 2015] Kim, Soo-Kyun; Liu, Fan; Choi, Sang-Soo. Multimedia applications for intelligent fusion computing. Multimedia Tools and Applications, 2015, vol. 74, no 10, p. 3273-3276.

[Thomson 2009] Thomson, Jim; Jessup, Andrew T. A Fourier-based method for the distribution of breaking crests from video observations. Journal of Atmospheric and Oceanic Technology, 2009, vol. 26, no 8, p. 1663-1671.

[USAL 2020] Universidad de Salamanca. Departamento de Geología. <http://hidrologia.usal.es/temas/Precipitaciones.pdf>
Fecha de revisión 2020.

[Velázquez 2014] Tesis de Jorge Eduardo Velázquez Suárez. Título: "Calibración de un modelo de estimación de lluvia con imágenes de satélite, utilizando datos de estaciones climatológicas, para la región hidrológica número 30 de México". México 2014.

[VIMETRI-MAC] Entidad financiadora: Programa de Cooperación Territorial. INTERREG V A España-Portugal. MAC 2014-2020. TÍTULO: "Sistema de

vigilancia meteorológica para el seguimiento de riesgos medioambientales”.
ACRÓNIMO: VIMETRI-MAC. CÓDIGO: MAC/3.5b/065.

[Webster 2015] Webster, Dereck D.; Breckon, Toby P. Improved raindrop detection using combined shape and saliency descriptors with scene context isolation. En 2015 IEEE International Conference on Image Processing (ICIP). IEEE, 2015. p. 4376-4380.

[Wikipedia 2020] Wikipedia. Ciclo hidrológico del agua. Disponible en:
https://es.wikipedia.org/wiki/Ciclo_hidrol%C3%B3gico
Fecha de revisión septiembre del 2020.

[Wikipedia 2021] Wikipedia. Filtro Sobel. Disponible en:
https://es.wikipedia.org/wiki/Operador_Sobel
Fecha de revisión Junio del 2021.

[Witten 2005] I. H. Witten y E. Frank, “Credibility: Evaluating What's Been Learned” de “Data Mining Practical Machine Learning Tools and Techniques”, Elsevier, 2005, pp. 143-185.

[Xueyang 2017] Fu, Xueyang, et al. Removing rain from single images via a deep detail network. En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. p. 3855-3863.

[Yao 1991] Wang, Yao; Zhu, Qin-Fan. Signal loss recovery in DCT-based image and video codecs. En Visual Communications and Image Processing'91: Visual Communication. International Society for Optics and Photonics, 1991. p. 667-678.

Parte II

Pliego de condiciones

Pliego de condiciones

En la segunda parte de esta memoria abordaremos la temática del pliego de condiciones, para ello, es necesario establecer dos puntos importantes indicados a continuación:

- **Pliego de condiciones de los recursos utilizados:** Se debe de indicar los recursos, tanto hardware como software, utilizados en el presente proyecto.
- **Pliego de condiciones legales y administrativas:** Se exponen aspectos legales y administrativos para el uso y explotación de este TFM.

Pliego de condiciones de los recursos utilizados

Recursos hardware

Para este TFM se han empleado los siguientes recursos tipo hardware:

- **Ordenador portátil:**
 - Microprocesador Intel Core i7 de octava generación (8750H) con una velocidad de CPU 2,21 GHz
 - Memoria Ram de 16 GB y DDR4.
 - Disco duro sólido de 256 GB.
 - Disco duro de 1 TB.
 - Tarjeta gráfica de Nvidia, GeForce y GTX1060.
- **Ordenador de sobremesa:**
 - Microprocesador Intel(R) Core™ i7-7700 CPU @ 3.60GHz 3.60 GHz

- Memoria Ram de 16 GB y DDR4.
- Disco duro sólido de 128 GB.
- Dos discos duros de 1 TB/unidad.
- Tarjeta gráfica de Nvidia, GeForce y GTX620.

- **Cámaras:**

En el capítulo 3 de este TFM se detallan los modelos y las características técnicas de las cámaras empleadas. Como recordatorio, se expone de nuevo, los modelos y sus características a continuación.

1. Modelo Grandstream GXV3674-FHD (Imagen 229).



Imagen 229. Cámara modelo Grandstream GXV364_FHD.

Las características técnicas de esta cámara se ilustran en la tabla 20:

Tabla 20. Características técnicas cámaras GXV364_FHD.

| | |
|--|---|
| Compresión de vídeo | H.264, MJPEG |
| Resolución del sensor de imagen | 1/3", 3.1 Megapixel Progressive Scan CMOS, 2048H x 1536V |
| Sensibilidad del sensor de imagen | Modo día/noche. Bajo ruido y baja sensibilidad a la luz. Obturador: 1/10000 - 1/30 segundo |
| Distancia Focal | 2.8 ~ 12 mm |
| Apertura | F1.4, DC-IRIS |
| Angulo de campo | 90° ~ 28° (Campo horizontal) |
| Filtro de corte IR | Si, mecánico |
| Modo día/noche | IR LED cubre hasta 25m |
| Iluminación mínima | 0.05 Lux, 0 lux con IR |
| Receptividad | 1.9V/lux-segundo (550nm) |

| | |
|---|---|
| Video máximo admitido | 2048x1536 (15 imágenes) |
| Resolución y velocidad de fotogramas | 1920x1080 (30 imágenes) |
| Velocidad bit | 32 Kbps ~ 8 Mbps velocidad múltiple para vista previa y grabación |
| Entrada de audio | Line-In, 1500 ohms, 0.1Vrms ~ 0.707Vrms, 3.5mm |
| Salida de audio | Line-Out, 600 ohms, 0.707Vrms, 3.5mm |
| Compresión de audio | G.711u/a, AAC |
| Analítica empotrada | Detección de movimiento (hasta 16 áreas objetivo) |
| Pre-/post-alarma de grabación | Si, 3MB |
| WDR (Wide Dynamic Range) | Si (100dB) |
| Instantáneas | Activado por eventos, enviar por correo electrónico / FTP |
| Protocolo de red | TCP/UDP/IP, RTP/RTCP, RTSP, DHCP, DDNS, HTTP, HTTPS, SMTP, FTP, NTP |
| Soporte SIP/VoIP | Si |
| Fotografía de lapso de tiempo | Si |
| Máscara de privacidad | Si, 4 zonas |
| Power over Ethernet (PoE) | IEEE 802.3af, Clase 0 |
| Conexión de cables externos | Red: RJ45, 10M/100M autodetección 3.5mm Line-In 3.5mm Line-Out Power Input |
| Dimensiones | 255mm (L) x 104mm (W) x 187mm (H) |
| Peso | 1.10kg |
| Temperatura/Humedad | En funcionamiento: -20°C ~ +50°C (-4°F ~ 122°F), 10–90% RH (no-condensación) En almacenamiento: -30°C ~ +60°C (-22°F ~ 140°F) |
| Adaptador de alimentación | Salida: 12VDC/1A; Entrada: 100–240VAC, 50–60Hz |
| Caja externa o carcasa | Carcasa de metal resistente a la intemperie IP66 |

| | |
|--------------------|--|
| Conformidad | FCC Parte 15, Subparte B Clase B; EN 55022 Clase B, EN 61000-3-2, EN 61000-3-3, EN 55024, EN 60950-1; C-tick AS/NZS CISPR 22, IP66 |
|--------------------|--|

1. Modelo FOSCAM FI9901EP (Imagen 230).



Imagen 230. Cámara modelo FOSCAM FI9901EP.

Las características técnicas de esta cámara se ilustran en la tabla 21:

Tabla 21. Características técnicas cámaras FOSCAM FI9901EP.

| | | |
|-------------------------|------------------|---|
| SENSOR DE IMAGEN | Sensor | High Definition Color 1/3" CMOS Sensor |
| | Resolución | 2560 x 1440 Pixels (4.0 Megapixels), 2304 x 1296 Pixels (3.0 Megapixels). 2X Magic Zoom and 6X Digital Zoom |
| | Iluminación | 0 Lux mínimo (con iluminación infrarroja incorporada) |
| | Controles | Control de brillo, contraste y frecuencia de luz. Automáticos y manuales |
| LENTES | Lente | Cristal; Lentes IR-infrarrojas de visión nocturna. f: 2.8 mm, 78° visión diagonal |
| VÍDEO | Compresión vídeo | H.264 |
| | Stream | Doble stream |
| | Imágenes/seg. | 30 fps máx. (ajustable a valores inferiores) |
| | Resolución | 2560 x 1440 Pixels (4.0 Megapixels), 2304 x 1296 Pixels (3.0 Megapixels). 2X Magic Zoom and 6X Digital Zoom |
| | Volteo imagen | Vertical / Horizontal |

| | | |
|----------------------------------|--|---|
| | Frecuencia luz | 50Hz, 60Hz o Exterior |
| | Modo infrarrojo | Automático y Manual |
| | Ajustes vídeo | Brillo, Contraste |
| SONIDO | Entrada/Salida | Conectores jack estándares 3.5 mm para entrada/salida de audio. Compresión PCM/G.726 |
| COMUNICACIÓN | Red Ethernet | 10/100 Mbps RJ-45 con PoE IEEE802.3af |
| | Protocolos | HTTP, P2P (acceso vía Internet sin necesidad de abrir puertos), FTP, TCP/IP, UDP, SMTP, DHCP, PPPoE, UPnP, GPRS |
| ALMACENAMIENTO | Tarjeta microSD | Ranura para tarjeta microSD |
| SISTEMAS COMPATIBLES | Sistema operativo | Windows XP, Vista, 7, 8, 10; MacOS, iOS, Android |
| | Navegador | IE, Firefox, Chrome, Safari |
| DATOS FÍSICOS | Luz infrarroja | IR LEDs, Alcance nocturno hasta 20 m |
| | Dimensiones | En mm.: 98 (La) x 65 (An) x 62 (Al) |
| | Orientación | 180° horizontal / 180° vertical |
| | Peso | 500 gr (accesorios incluidos) |
| ALIMENTACIÓN | Alimentación | Adaptador DC 12V/1.0A (incluido). El cable mide 1,5 m. |
| | PoE | Alimentable eléctricamente por PoE (cable ethernet). Estándar IEEE802.3af |
| | Consumo | <5 W |
| ENTORNO | Temperatura | -20° ~ 55°C (operativa) -20°C ~ 60°(almacenamiento) |
| | Humedad | 10% ~ 80% sin condensación (óptima operativa) 0% ~ 90% sin condensación (almacenamiento) |
| CERTIFICACIÓN Y NORMATIVA | CE, RoHs, FCC, IP66 (estanqueidad en modelos exterior) | |

- **Equipamiento de cámaras:**

- Herraje de instalación
- Mástil de sujeción
- Cableado
- Armario de exterior con ventilación formado por:
 - Un Switch 8 puertos con POE UBNT-ES-8-150W.
 - Un PSU / Rectificador 48-56V.
 - Un Cuadro eléctrico con magnetotérmico 2P 16A, diferencial 2P 25A 30mA y monitor de tensión con contacto NNNC.
 - Con su correspondiente herraje de instalación, mástil de sujeción y cableado.

- **Equipamiento de recepción de los datos en el laboratorio:**

- Un Router Mikrotik RB2011
- Un Sistema de alimentación ininterrumpida (SAI) de 700VA marca Salicru SPS.900.ONE

- **Pluviómetros:**

En el capítulo 2 de este TFM se detallan los modelos y las características técnicas de los pluviómetros empleados. Como recordatorio, se expone de nuevo, los modelos y sus características a continuación.

1. Modelo VAISALA WXT530 (Imagen 231).



Imagen 231. Pluviómetro modelo WXT530.

Las características técnicas de este pluviómetro se ilustran en la tabla 22.

Tabla 22. Características técnicas del pluviómetro VAISALA WXT530.

| | |
|--|---|
| Área de captación | 60cm ² . |
| Precipitación | Por acumulación después del último reinicio automático o manual. |
| Resolución | 0.01mm. |
| Precisión para la acumulación diaria. | Mejor al 5%. |
| Duración | Incrementos de 10 segundos desde que se detecta la precipitación. |
| Intensidad | La ejecución tiene un promedio de 1 minuto para pasos de 10 segundos. |
| Rango de la intensidad | 0-200mm. |
| Resolución de la intensidad | 0.1mm/h. |
| Granizo | Cantidad de golpes acumulados contra la superficie de captación. |

2. Modelo VAISALA RG13 (Imagen 232).



Imagen 232. Pluviómetro modelo RG13.

Las características técnicas de este pluviómetro se ilustran en la tabla 23:

Tabla 23. Características técnicas del pluviómetro modelo VAISALA WXT530.

| | |
|-----------------------------------|----------------------|
| Precisión | Mejor al 2%. |
| Diámetro de apertura | 225mm. |
| Área de apertura | 400cm ² . |
| Capacidad de precipitación | Ilimitada |
| Resolución | 0.2mm. |
| Duración | <100ms |

Recursos software

Para este TFM se han empleado los siguientes recursos tipo software:

- **MATLAB versión 2019b:** programa empleado para el desarrollo de los algoritmos de este TFM. Para la correcta ejecución de este TFM, el matlab debe tener incluidas las librerías:
 - Bioinformatics Toolbox
 - Computer Vision Toolbox
 - Deep Learning Toolbox
 - Image Acquisition Toolbox
 - Image Processing Toolbox
 - Optimization Toolbox
 - Reinforcement Learning Toolbox
 - Signal Processing Toolbox
 - Statistics and Machine Learning Toolbox
 - Vision HDL Toolbox
 - Wavelet Toolbox
 - Audio Toolbox
 - Econometrics Toolbox
 - Financial Toolbox
 - Fuzzy Logic Toolbox
 - Data Base
 - Mapping
 - Matlab compiler (ambos)

- **Windows versión 10:** Sistema operativo empleado.
- **Microsoft Office versión 365:** Para la elaboración de la memoria, gráficos e ilustraciones se emplearon diferentes paquetes de herramientas de Microsoft Office. Tales como Microsoft Word, Power Point, entre otros.
- **WordPad predeterminado de Windows 10:** Programa que permite la apertura y edición de textos de los algoritmos de MATLAB.
- **Base de datos:** Los datos meteorológicos se han obtenido de una plataforma web gestionada por el Proyecto VIMETRIC-MAC. Los datos recogidos en este TFM han sido autorizados por los responsables de este proyecto (Navarro Mesa y Ravelo García).

Pliego de condiciones legales y administrativas

En este apartado, se exponen aspectos legales y administrativos para el uso y explotación de este TFM. A continuación, se detallarán los derechos, deberes y obligaciones ante el uso de los algoritmos desarrollados en este TFM por futuros usuarios con los autores.

Derechos de explotación y restricciones del uso de los algoritmos de este TFM

La Universidad de Las Palmas de Gran Canaria es quien ostenta los derechos de explotación y cualquier usuario debe de estar de acuerdo en obligarse por los términos y condiciones establecidas en este Pliego de Condiciones, aceptando todos sus términos y estipulaciones; de igual manera se deberá de cumplir lo

establecido en el “Reglamento general de los trabajos de fin de grado y de fin de máster de las titulaciones oficiales de la Universidad de Las Palmas de Gran Canaria”, aprobado por acuerdo de Consejo de Gobierno de la ULPGC de 28 de noviembre de 2019 (BOULPGC 5 diciembre 2019), que en su capítulo VI reglamenta los derechos de propiedad intelectual, de explotación industrial y protección de datos personales.

El uso de los programas, ya sean completos o de forma parcial, base de datos o una copia desde el ordenador personal de los mismos, será bajo la autorización expresa del autor, los tutores del proyecto y la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria.

El usuario no podrá realizar ingeniería inversa, decompilación o desensamblado de los programas o partes de estos. El usuario podrá transferir los programas y bases de datos a un tercero, siempre que no tenga copias de los mismos, incluyendo posibles actualizaciones o retener material escrito adicional que acompañe a los programas y la base de datos.

Garantías y consideraciones adicionales

La autora y los tutores de este TFM garantizan que los programas o partes de estos, y la base de datos funcionarán correctamente en el momento de la correcta instalación. También se garantiza que el soporte en el cual está grabado el programa no contendrá defectos en el momento de la adquisición del mismo.

Con la única excepción de lo expresamente expuesto en el párrafo anterior, los programas y bases de datos han sido creados sin garantías de ningún tipo. No se asegura, garantiza o realiza ninguna declaración respecto al uso o los resultados derivados de la utilización de los programas o la base de datos. Tampoco se garantiza que la operación de los programas sea ininterrumpida o sin errores.

Limitación de responsabilidad

En ningún caso serán la autora ni los tutores del TFM, ni la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria responsables de los perjuicios directos, indirectos incidentales o consiguientes, gastos, lucro cesante, pérdida de ahorros, interrupción de negocios, pérdida de información comercial o de negocio, o cualquier otra pérdida que resulte del uso o de la incapacidad de usar los programas, la base de datos o la documentación. El usuario conoce y acepta esta asignación de riesgo, así como el resto de cláusulas, garantías, derechos, deberes y restricciones. La autora y los tutores rechazan cualquier otras cláusulas, garantías, derechos, deberes y restricciones que no hayan sido indicadas anteriormente en este Pliego de Condiciones.

Varios

Las cláusulas, garantías, derechos, deberes y restricciones expuestas en el Pliego de condiciones se rigen por las leyes del estado español. El usuario acepta la jurisdicción exclusiva de los tribunales de este país en relación con cualquier disputa que pudiera derivarse de las cláusulas, garantías, derechos, deberes y restricciones de este TFM.

Parte III

Presupuesto

Presupuesto

Doña Naira Dolores Rosales Hernández, autora del presente Proyecto de Fin de Máster, declara que:

El proyecto Fin de Máster con título “Técnicas de aprendizaje máquina para la detección y cuantificación de lluvia sobre imágenes de videovigilancia”, desarrollado en la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria y en un periodo de 4 meses, tiene un coste total de dieciséis mil quinientos setenta y siete euros con sesenta y seis céntimos (16.577,66€) correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

Firmando la presente para que así conste a los efectos oportunos.

Autora del Proyecto:

Naira Dolores Rosales Hernández

Las Palmas de Gran Canaria, 9 de julio de 2021

Desglose del presupuesto

En este apartado se procederá a describir y desglosar los gastos que han generado el TFM. En dicho desglose se han empleado precios orientativos sugeridos por el Colegio Oficial de Ingenieros de Telecomunicación (COIT) y de la Asociación Española de Ingenieros de Telecomunicación (AEIT), así como también precios consultados actuales del mercado.

El presupuesto se desglosa en 6 partes para la estimación y cálculo de los gastos generados en el TFM:

- Trabajo tarifado por tiempo empleado.
- Amortización del material instrumental, recursos hardware y software.
- Redacción del TFM.
- Derechos de visado del COIT (opcional).
- Gastos de tramitación y envío.
- Aplicación de impuestos.

Trabajo tarifado por tiempo empleado

En este apartado se procederá a realizar los cálculos de los gastos que derivan de la remuneración por hora de un ingeniero Graduado en Ingeniería en Tecnologías de la Telecomunicación. El cálculo estimado de los honorarios no está establecido en ningún documento o reglamento, se establece de manera libre por el ingeniero autor del proyecto. Por ello, se ha procedido a realizar la siguiente ecuación para el cálculo del mismo (Ecuación 18).

$$H = 20 * H_N + 30 * H_E \text{ euros}$$

Ecuación 18. Cálculo de los honorarios.

En donde:

- H son los honorarios por tiempo.
- H_N son las horas trabajadas dentro de la jornada laboral. Se ha multiplicado por 20 porque es el precio en euros/hora en horario laboral.

- H_E son las horas especiales trabajadas. Se ha multiplicado por 30 porque es el precio en euros/hora en horario fuera del horario laboral.

Para la realización del presente TFM, se estima que se ha trabajado durante 4 meses. El tiempo empleado por día consta de 6 horas diarias. A demás, un mes tiene aproximadamente 20 días laborables, por ello, el cálculo de las horas trabajadas dentro de la jornada laboral es el siguiente (Ecuación 19).

$$H_N = 4 \times 20 \times 6 = 480 \text{ horas}$$

Ecuación 19. Cálculo de las horas trabajadas dentro de la jornada laboral.

Para el cálculo de las horas realizadas fuera de la jornada laboral, es decir, los fines de semana es la siguiente (Ecuación 20). Durante los 4 meses, todos los fines de semana se ha empleado un total de 6 horas, y cada mes tiene aproximadamente 4 fines de semana.

$$H_E = 4 \times 4 \times 6 = 96 \text{ horas}$$

Ecuación 20. Cálculo de las horas trabajadas dentro de la jornada laboral.

Por lo tanto, los honorarios asociados al total de horas trabajadas en este TFM han sido (Ecuación 21):

$$H = 20 * 480 + 30 * 96 = 12.480 \text{ €}$$

Ecuación 21. Cálculo de los honorarios.

Los honorarios totales por tiempo dedicado, libres de impuestos, ascienden a doce mil cuatrocientos ochenta euros (12.480€).

Amortización del material instrumental y recursos Hardware/Software.

En este apartado se procederá a realizar los cálculos de amortización de los materiales de equipamiento e instrumental y los recursos hardware y software necesarios para el desarrollo de este TFM. Para escoger un porcentaje o coeficiente de amortización lineal haremos uso de las referencias estimadas de las tablas de coeficientes de amortización lineal de la Agencia Tributaria (AEAT).

Los recursos materiales empleados en este TFM son los siguientes:

- **Recursos software:** Programa MATLAB para el desarrollo de algoritmos, paquete de herramientas de Microsoft Office para la elaboración de la memoria, gráficos e ilustraciones, el sistema operativo Windows para la elaboración y ejecución de este TFM y programa WordPad para la apertura y edición de textos de los algoritmos de MATLAB.
- **Recursos hardware:** Recursos tales como el ordenador de sobremesa, monitor, ordenador portátil, pluviómetros, cámaras, etc.

Amortización de los recursos hardware

En la siguiente tabla se ha detallado la cuota de amortización anual de cada uno de los recursos materiales hardware empleados (Tabla 24).

Tabla 24. Precios y costes de amortización de recursos hardware.

| Elementos | Cantidad | Valor unidad (€) | Coste total (€) | Coefficiente lineal (%) | Coste de amortización (€) |
|---|----------|------------------|-----------------|-------------------------|---------------------------|
| Ordenador portátil Intel Core i7 | 1 | 1.400 | 1.400 | 25 | 350 |
| Ordenador sobremesa Intel Core i7 | 1 | 1.100 | 1.100 | 25 | 275 |
| Monitor ACER 28" | 1 | 190 | 190 | 20 | 38 |
| Cámara Grandstream GXV3674-FHD | 6 | 220 | 1.320 | 20 | 264 |
| Cámara FOSCAM FI9901EP | 2 | 170 | 340 | 20 | 68 |
| VAISALA WXT530 | 1 | 1.000 | 1.000 | 20 | 200 |
| VAISALA RG13 | 1 | 800 | 800 | 20 | 160 |
| Total | | | | | 1.355 |

El coste total de amortización de los recursos hardware usados en este TFM ascienden a la cantidad de mil trescientos cincuenta y cinco euros (1.355 €).

Amortización de los recursos software

En la siguiente tabla se ha detallado la cuota de amortización de los recursos softwares empleados (Tabla 25):

Tabla 25. Precios y costes de amortización de recursos software.

| Elementos Software | valor unidad (€) | Coefficiente lineal (%) | Coste de amortización (€) |
|---|------------------------|-------------------------------|---------------------------------|
| MATLAB 2019b | 2.500 | 33 | 825 |
| Sistema Operativo Microsoft Windows 10 | 135 | 33 | 44,55 |
| Paquete Microsoft Office® 365 | 153,99 | 33 | 50,82 |
| WordPad | 0 | 0 | 0 |
| Total | | | 920,37 |

El coste total de amortización de los recursos software usados en este TFM ascienden a la cantidad de novecientos veinte euros con treinta y siete céntimos (920,37€). Los costes que aparecen a cero de WordPad es porque Microsoft no quiere monetizar su uso en la actualidad.

Amortización total del material instrumental y recursos Hardware/Software.

Una vez calculados los costes de recursos software y hardware, podemos establecer el coste total de amortización del material instrumental y recursos Hardware/Software (Tabla 26).

Tabla 26. Precios y costes de amortización de recursos software y software total.

| Elemento | Coste (€) |
|-------------------|----------------|
| Recursos Hardware | 1.355 |
| Recursos Software | 920,37 |
| TOTAL | 2275,37 |

El coste total de amortización de los recursos software y hardware usados en este TFM ascienden a la cantidad de dos mil doscientos setenta y cinco euros con treinta y siete céntimos (2275,37€).

Cuotas de redacción

El coste asociado a la redacción del proyecto se ha calculado según la siguiente expresión (Ecuación 22):

$$R = 0,05 * P$$

Ecuación 22. Coste de redacción.

Donde:

- R son los honorarios por la redacción del TFM.
- P es el presupuesto del proyecto obtenido.

Dicho valor se ha calculado sumando los costes de las secciones anteriores correspondientes a la amortización del inmovilizado material, tanto hardware como software y al trabajo tarifado por tiempo empleado.

Así, el valor de P asociado a este proyecto ha sido (Tabla 27).

Tabla 27. Precios y costes de amortización de recursos software y software total y trabajo tarifado por tiempo empleado.

| Concepto | Coste (€) |
|---------------------------------------|------------------|
| Trabajo tarifado por tiempo empleado | 12.480 |
| Amortización de los recursos hardware | 1.355 |
| Amortización de los recursos software | 920,37 |
| Total | 14.755,37 |

Teniendo en cuenta el presupuesto calculado en la tabla anterior, se tiene que:

$$R = 0,05 * P = 0,05 * 14.755,37 = 737,77 \text{ €}$$

Por lo tanto, el importe final de redacción del proyecto asciende a la cantidad de setecientos treinta y siete euros con setenta y siete céntimos (737,77€).

Aplicación de impuestos a los costes totales

Sumando todos los conceptos calculados hasta el momento, se obtiene el total del presupuesto previo a la aplicación de impuestos, como se muestra a continuación (Tabla 28).

Tabla 28. Aplicación de impuestos a los costes totales.

| Concepto | Coste (€) |
|---------------------------------------|------------------|
| Trabajo tarifado por tiempo empleado | 12.480 |
| Amortización de los recursos hardware | 1.355 |
| Amortización de los recursos software | 920,37 |
| Coste redacción del proyecto | 737,77 |
| Total antes de impuestos | 15.493,14 |
| Aplicación de Impuestos (7% IGIC) | 1084,52 |
| Total presupuesto | 16.577,66 |

El proyecto Fin de Máster con título “Técnicas de aprendizaje máquina para la detección y cuantificación de lluvia sobre imágenes de videovigilancia”, desarrollado en la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria y en un periodo de 4 meses, tiene un coste total de dieciséis mil quinientos setenta y siete euros con sesenta y seis céntimos (16.577,66€) correspondiente a la suma de las cantidades consignadas a los apartados anteriores.

Las Palmas de Gran Canaria, a 9 de Julio de 2021

Fdo: Naira Dolores Rosales Hernández