

Método para la Detección de Cráteres Lunares  
utilizando herramientas de Aprendizaje  
Automático

Jesús Cabrera González

# Índice general

<b>1. Introducción</b>	<b>10</b>
1.1. Contribuciones y Publicaciones . . . . .	11
1.2. Organización de la Tesis . . . . .	12
<b>2. Estado del Arte</b>	<b>13</b>
2.1. Detección de Cráteres Lunares utilizando Procesamiento de Imágenes . . . . .	14
2.2. Detección de Cráteres Lunares utilizando Métodos de Aprendizaje Supervisado . . . . .	21
2.3. Discusión del Estado del Arte . . . . .	26
<b>3. Marco Teórico</b>	<b>28</b>
3.1. Cráteres Lunares . . . . .	30
3.2. Estimando la edad de la superficie . . . . .	31
3.3. Procesamiento de Imágenes . . . . .	34
3.3.1. Imagen Integral . . . . .	35
3.3.2. Plantillas de Convolución . . . . .	37

<i>ÍNDICE GENERAL</i>	2
3.3.3. Plantillas Haar . . . . .	39
3.4. Aprendizaje Automático . . . . .	39
3.4.1. Clasificación Binaria . . . . .	41
3.4.2. Adaboost . . . . .	42
3.4.3. SVM ( <i>Support Vectors Machines</i> ) . . . . .	45
<b>4. Metodología</b>	<b>47</b>
4.1. Preparación de las Muestras de Entrenamiento . . . . .	48
4.2. Preprocesamiento de las Muestras . . . . .	51
4.3. Extracción de Características . . . . .	52
4.4. Entrenamiento del Modelo . . . . .	58
4.5. <i>Support Vector Machines</i> . . . . .	58
4.6. Filtro de Dirección de la Luz . . . . .	60
4.6.1. Calculando la Dirección de la Luz . . . . .	61
4.7. Filtro de Uniformidad de la Subventana . . . . .	63
4.8. Filtrando las Muestras por Grados . . . . .	65
<b>5. Experimentos y Resultados</b>	<b>68</b>
5.1. Validación Cruzada de $k$ Iteraciones . . . . .	69
5.2. Sensibilidad y Especificidad . . . . .	70
5.3. Evaluación de los algoritmos . . . . .	71
5.3.1. Resultados de la Validación Cruzada . . . . .	71

<i>ÍNDICE GENERAL</i>	3
5.3.2. Método para Detección de Cráteres Lunares . . . . .	82
<b>6. Conclusiones</b>	<b>100</b>
6.1. Conclusiones . . . . .	100
6.2. Trabajo Futuro . . . . .	102
<b>Bibliografía</b>	<b>104</b>

# Índice de figuras

2.1.	Algoritmo propuesto por Saraiva. (Tomado de [2]) . . . . .	15
2.2.	Pasos para la obtención de bordes. En a) se obtiene una detección de los bordes utilizando morfología, en b) se realiza la obtención de bordes con el algoritmo de Canny y c) es el mapa de bordes obtenido. (Tomado de [18]) . . . . .	16
2.3.	Comparación de los resultados obtenidos en [15]. Las dos imágenes de arriba representan el reconocimiento aplicando el <i>Canny adaptivo</i> y las inferiores con un <i>Canny normal</i> . . . . .	20
2.4.	Creación de etiquetas de grupos de píxeles. (Tomado de [39]) .	23
2.5.	Volumen probabilístico obtenido de la FFT. (Tomado de [2]) .	25
3.1.	Foto lunar fotografiada por Tony Spacey. . . . .	29
3.2.	Cráteres presente en la superficie lunar. Imagen obtenida de la misión Apollo [1]. . . . .	30
3.3.	Proceso de formación de un cráter, se impacta el proyectil y posteriormente la onda de choque deforma los bordes y expande el diámetro del cráter [24]. . . . .	31
3.4.	Gráfica de distribución frecuencia-tamaño. Neukum et al [27] .	33
3.5.	Rectángulo que representa la imagen integral en el punto (x,y).	36

3.6. El valor de la imagen integral en la ubicación 1 es la suma de los píxeles comprendidos en el rectángulo A, es decir  $ii(1)=A$ . Análogamente,  $ii(2)=A+B$ ,  $ii(3)=A+C$ ,  $ii(4)=A+B+C+D$ . De esto se deduce que la suma de los píxeles comprendidos en D es  $ii(4)+ii(1)-[ii(2)+ii(3)]$ , para lo cual se requieren las 3 operaciones aritméticas mencionadas (dos sumas y una resta). . . . . 37

3.7. Ejemplo de aplicación de una plantilla de convolución. . . . . 38

3.8. Plantillas Haar: una de configuración vertical, otra horizontal, una diagonal y la última de 3 rectángulos. . . . . 39

3.9. Plantilla de convolución. Los valores de intensidad son restados según la configuración donde el negro representa el -1 y blanco 1. . . . . 40

3.10. Ejemplo de Clasificador binario. En esta imagen observamos que se dividen las instancias en dos clases (estrellas y triángulos, el modelo que clasifica las muestras es la ecuación de la línea curva dando lugar a que las muestras encima del objeto sean estrellas y las de abajo triángulos. . . . . 42

3.11. Ejemplo de la iteración para la elección de los clasificadores débiles en las iteraciones 1,2,3,4,5,6,7 y 8 que es el clasificador final o clasificador fuerte. . . . . 45

4.1. Proceso general empleado para la detección de cráteres. . . . . 48

4.2. Imagen fuente 1. . . . . 49

4.3. Imagen fuente 2. . . . . 49

4.4. Cráter original con 4 rotaciones. . . . . 50

4.5. Algunos ejemplos de las muestras que se utilizaron para el entrenamiento. . . . . 51

4.6. Coordenadas de la cuadrilla de 2x2. . . . . 53

<i>ÍNDICE DE FIGURAS</i>	6
4.7. Coordenadas de la cuadrilla de 3x3. . . . .	54
4.8. Configuraciones para subventanas de 2x2. . . . .	55
4.9. Configuraciones para subventanas de 3x3. . . . .	56
4.10. Aplicando las distintas plantillas en una muestra de entrena- miento. . . . .	57
4.11. Conversión de datos a un espacio de características. . . . .	60
4.12. Calculo de la dirección de la luz de un cráter. . . . .	62
4.13. El cálculo de uniformidad permite descartar muestras. . . . .	65
4.14. Umbral de 24 grados establecido para filtrar el resto de las muestras. . . . .	65
4.15. Histograma de dirección de la luz de un grupo de muestras detectadas por el clasificador en cuestión. . . . .	66
5.1. Gráfica del decremento del error al ir aumentando el número de clasificadores. . . . .	74
5.2. Gráfica del decrecimiento del error al ir aumentando el núme- ro de clasificadores. Observamos que aun teniendo un mejor desempeño, dos de los grupos tuvieron más errores. . . . .	78
5.3. Ejemplos de muestras catalogadas erróneamente. En la fila su- perior observamos “Falsos Positivos” y en fila inferior “Falsos Negativos”. . . . .	79
5.4. Imagen de la superficie lunar, extraída del banco de imágenes de la misión Apollo. . . . .	86
5.5. Etiquetado manual de cráteres presentes en la imagen de prue- ba 1. . . . .	87
5.6. Detección de cráteres en la imagen 1 utilizando “ <i>Adaptive Boosting</i> ” con un filtrado rápido. . . . .	88

5.7. Detección de cráteres en la imagen 1 utilizando “ <i>Adaptive Boosting</i> ” con el filtrado automático propuesto. . . . .	89
5.8. Detección de cráteres en la imagen 1 utilizando SVM con un filtrado rápido. . . . .	90
5.9. Detección de cráteres en la imagen 1 utilizando SVM con el filtrado automático propuesto . . . . .	91
5.10. Imagen del Mar de la Humedad (Imagen de prueba 2). . . . .	93
5.11. Etiquetado manual de cráteres presentes en la imagen de prueba 2. . . . .	94
5.12. Detección de cráteres en la imagen 2 utilizando “ <i>Adaptive Boosting</i> ” con un filtrado rápido. . . . .	95
5.13. Detección de cráteres en la imagen 2 utilizando “ <i>Adaptive Boosting</i> ” con el filtrado automático propuesto. . . . .	96
5.14. Detección de cráteres en la imagen 2 utilizando SVM con un filtrado rápido. . . . .	97
5.15. Detección de cráteres en la imagen 2 utilizando SVM con el filtrado automático propuesto. . . . .	98

# Índice de tablas

5.1.	Desempeño de los vectores de características generados al aplicar plantillas de [23]. Evaluando detecciones Negativas erróneas.	72
5.2.	Desempeño de los vectores de características generados al aplicar plantillas de [23]. Evaluando detecciones Positivas erróneas.	73
5.3.	Sensibilidad y Especificidad para Clasificador entrenado con características extraídas de plantillas genéricas. . . . .	73
5.4.	Desempeño de los vectores de características generados al aplicar plantillas propuestas. Evaluando detecciones Positivas erróneas.	75
5.5.	Desempeño de los vectores de características generados al aplicar plantillas propuestas. Evaluando detecciones Negativas erróneas.	76
5.6.	Sensibilidad y Especificidad para Clasificador entrenado con características propuestas. . . . .	76
5.7.	Tasa de detección alcanzada por el clasificador entrenado con Adaptive Boosting para cada una de las particiones. . . . .	77
5.8.	Desempeño del clasificador <i>SVM</i> con un kernel gaussiano radial.	80
5.9.	Sensibilidad y Especificidad para Clasificador entrenado con características propuestas. . . . .	80
5.10.	Desempeño del clasificador <i>SVM</i> con un kernel polinomial de orden 3. . . . .	81

<i>ÍNDICE DE TABLAS</i>	9
5.11. Sensibilidad y Especificidad para Clasificador entrenado con características propuestas. . . . .	81
5.12. Desempeño del clasificador entrenado por SVM en cada una de las particiones. . . . .	82
5.13. Comparación de los cráteres detectados y los cráteres etiquetados manualmente para la imagen 1. El filtrado 1 corresponde al filtrado “rápido” y el 2 es el filtrado automático propuesto.	92
5.14. Comparación de los cráteres detectados y los cráteres etiquetados manualmente para la imagen 2. El filtrado 1 corresponde al filtrado “rápido” y el 2 es el filtrado automático propuesto.	99

# Capítulo 1

## Introducción

La astronomía es una ciencia que necesita de herramientas para poder analizar y comprender los fenómenos que suceden en astros que conforman el universo, concretamente aquellos astros que han sido explorados por sondas, telescopios entre otras herramientas utilizadas para la exploración espacial. Para realizar la investigación de los fenómenos, se utilizan herramientas de computación como lo son algoritmos de reconocimiento, algoritmos de procesamiento de imágenes, algoritmos de simulación numérica entre otros. Estos han sido de gran relevancia para el análisis de imágenes, y su estudio permite adquirir de manera más óptima datos e información. La astronomía es una de las ciencias que necesitan un apoyo visual de los datos obtenidos, es por ello que los algoritmos de procesamiento de imágenes son óptimos para la extracción de información, los cuales son apoyados por los de aprendizaje automático para el reconocimiento e identificación de esta información.

Una de las cosas importantes que estudia la astronomía son las superficies de los astros, lo cual es posible debido a los avances tecnológicos, y al alcance que tiene ahora sondas espaciales que fueron enviadas tiempo atrás. Estudiar la superficie del astro permite conocer muchas cosas de él, como su evolución geológica, o de igual manera para identificar y caracterizar los objetos que son responsables de los cráteres presentes en él. Estudiar estas formaciones topográficas, como otras de distinta clase permite conocer la historia y evolución geológica del astro, así como un seguimiento del origen de los objetos que impactan en ellos, es por eso que cobra relevancia la impor-

tancia de la detección de estos ya que una vez detectados, se recurre a la interpretación del tamaño de los cráteres, así como la distribución que tienen estos a lo largo y ancho de la superficie que se encuentra en estudio. Conocer las características de los cráteres nos permite entender la historia del astro.

Actualmente existen diversas herramientas que hacen posible la detección de cráteres que detallaremos en el presente documento, de igual manera podemos observar métodos que realizan una detección más eficiente que otros métodos que encontramos, ya que siguen otros modelos que fueron previamente entrenados. Existen múltiples enfoques en donde cada uno posee sus ventajas y desventajas, siendo de manera general que estos métodos utilizan métodos donde se replican el uso de plantillas para extraer características, y en otros se siguen métodos donde se combinan el uso de varios algoritmos, lo cual es un problema muy común en el aprendizaje automático a la hora de asignar etiquetas al realizar la clasificación de las muestras. Entre estos métodos encontramos los que utilizan máquinas de soporte vectorial *SVM* y los de *Adaptive Boosting* en donde se utilizan distintos tipos de características siendo las de tipo Haar relevantes para el entrenamiento de *Adaptive Boosting*.

Una de las problemáticas más importantes es que las plantillas deben ser definidas empíricamente de acuerdo al objeto de estudio, es por ello que en este caso se sugieren un grupo de plantillas que sumadas a las tradicionales servirán para la obtención de un clasificador para la primera etapa del método. De igual manera este trabajo se enfoca en la construcción de un sistema de filtros para la detección de cráteres lunares, utilizando un sistema de cálculo de luminosidad y filtros de imágenes especializados para este problema en específico, que servirá para la eliminación de muestras detectadas erróneamente. De esa manera se espera poder obtener un clasificador que funcione de mejor manera para imágenes en general que contengan cráteres lunares.

## 1.1. Contribuciones y Publicaciones

En este trabajo se logró contribuir con lo siguiente:

1. Un método de clasificación cráteres lo cual es de gran relevancia para el estudio de astros, así como para el posterior análisis de los cráteres detectados.
2. Un sistema de filtrado de imágenes por dirección de la luz, lo cual es relevante para ciertos problemas de procesamiento de imágenes.
3. Una base de datos de 960 muestras positivas y 960 muestras negativas de cráteres lunares, lo cual ayudará a la posterior evaluación de otros algoritmos.
4. La publicación del artículo *Towards an automatic counter of lunar craters* presentado en el congreso de Electrical Engineer, Computing Science and Automatic Control (CCE) en Septiembre de 2014.

## 1.2. Organización de la Tesis

Este trabajo se desarrolla como sigue: en el capítulo 2 encontraremos la revisión de los trabajos previos que nos llevaron a formular la solución propuesta, son trabajos que revisamos de las dos áreas que aportan al método de este trabajo. En el capítulo 3 encontraremos el marco teórico del problema en sí. En el capítulo 4 se detallará la metodología propuesta y cada uno de los pasos a seguir. En el capítulo 5 se detallarán los experimentos realizados, de igual manera se analizarán los resultados obtenidos en las diversas pruebas que se realizaron con el método propuesto. Por último en el capítulo 6 se cierra el trabajo con las conclusiones de lo que se consiguió en esta tesis, así como las propuestas y el trabajo futuro.

# Capítulo 2

## Estado del Arte

El problema de la detección de cráteres ha sido abordado desde múltiples enfoques; podemos encontrar enfoques que utilizan herramientas orientadas hacia la búsqueda de elementos elípticos o circulares, los cuales buscan dichos elementos largo de los bordes de los cráteres utilizando la transformada de Hough y métodos de preprocesamiento de imágenes. Por otro lado encontramos trabajos enfocados al reconocimiento de patrones en imágenes que nos llevan a entrenar un clasificador que se encarga de clasificar un conjunto de píxeles, catalogándolo como cráteres o no cráter; en dichos trabajos se utilizaron técnicas de aprendizaje automático utilizadas para la detección de los cráteres en superficies lunares.

En esta sección se presenta un resumen de algunos trabajos previos que podemos encontrar que abordar el problema de detección de cráteres, tanto los que tienen un enfoque de procesamiento de imágenes como los que utilizan alguna herramienta de aprendizaje automático.

## 2.1. Detección de Cráteres Lunares utilizando Procesamiento de Imágenes

El número de trabajos que abordan el problema de detección de cráteres utilizando herramientas de procesamiento de imágenes es mayor a los que tienen un enfoque orientado hacia el aprendizaje máquina. Podemos observar que una de las características que presentan estos trabajos es que se realiza un pre procesamiento a la imagen para de esa manera poder tener la información de los bordes la cual es de relevancia para realizar el ajuste de elipses, la transformada de Hough, entre otros[9],[41].

Como primer trabajo podemos observar el trabajo de [2], donde la fase inicial hace énfasis en la detección de bordes siguiendo de un proceso donde se realiza un cálculo de volumen probabilístico con distintos templates para poder catalogar un grupo de píxeles como cráter o no cráter. Este procedimiento ocurre en 3 etapas:

1. Detección de bordes: en la detección de bordes se hace un énfasis en la identificación de los bordes de los cráteres, obteniendo una imagen binaria que contiene la ubicación de los bordes.
2. Template matching: posteriormente realizaron un correlacionamiento entre la imagen y la medición calculada de similitud entre ellos, simplemente utilizaron templates con la forma de un círculo con un anillo exterior brillante y un centro oscuro, posteriormente se utilizó la transformada rápida de Fourier para el cálculo de la correlación.
3. Por último, se realiza una búsqueda probabilística para descartar firmas no identificables siguiendo ciertas pautas morfológicas como lo son la dimensión o la redondez. Terminando el proceso crítico obtenemos los probables centros de los cráteres identificados.

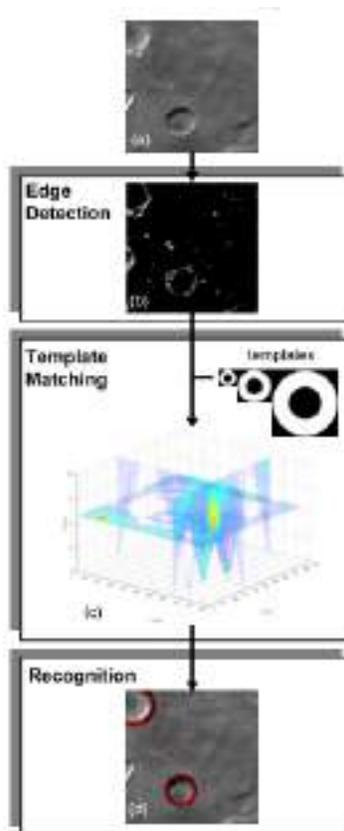


Figura 2.1: Algoritmo propuesto por Saraiva. (Tomado de [2])

El método discutido anteriormente posee una baja tasa de detección (78%), lo cual nos lleva a otro enfoque [18], en donde realizan un procesamiento de la imagen para poder obtener la apertura y cierre morfológico, y de esa manera obtener un mapa de bordes necesario para realizar el ajuste de elipses [9]. Este procedimiento es posible gracias a que una imagen típica de un cráter contiene el borde circular encerrando un patrón de zonas oscuras y brillantes de acuerdo al ángulo del sol, así como la topografía del cráter; las cuales son características muy utilizadas para la detección de cráteres.

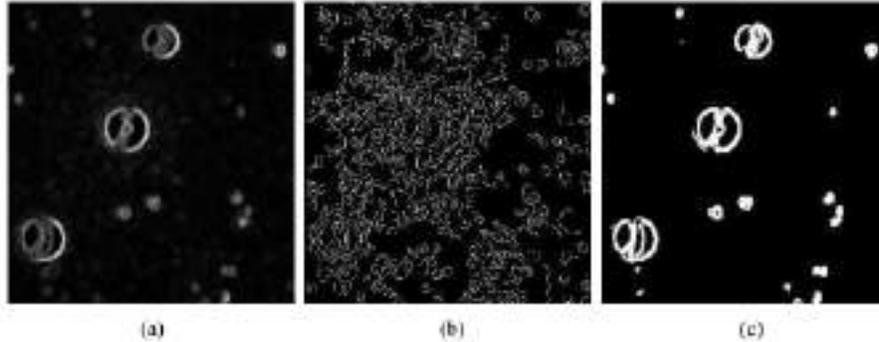


Figura 2.2: Pasos para la obtención de bordes. En a) se obtiene una detección de los bordes utilizando morfología, en b) se realiza la obtención de bordes con el algoritmo de Canny y c) es el mapa de bordes obtenido. (Tomado de [18])

El método de [18], sigue una serie de pasos morfológicos para obtener un mapa binarizado en el cual se encuentran representados los bordes. Estos pasos se encargan de englobar la parte brillante, así como la oscura, para posteriormente unir ambos lados formando elementos circulares (de ahí que los gradientes morfológicos aplicados tienen una forma circular). Antes de proseguir con el ajuste de elipses, se aplica una umbralización para separar los bordes detectados del fondo.

Posteriormente, el método de ajuste de elipses que se utiliza para la detección de bordes, es usado mediante una aproximación de análisis de regresión llamada mínimos cuadrados; la cual realiza un ajuste para tener una ecuación que se ajuste a los datos teniendo el error mínimo cuadrático. Este método al contrario que otros como la transformada de Hough requiere menos tiempo y es menos complejo. El ajuste de elipses es realizado con las siguientes ecuaciones:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (2.1)$$

Donde los pixeles candidatos son sustituidos en la ecuación paramétrica y de esta manera se calculan los parámetros de la elipse. La orientación es

calculada utilizando la siguiente ecuación:

$$\theta = \frac{1}{2} \tan^{-1} \frac{B}{C - A} \quad (2.2)$$

Para posteriormente graficar la elipse encerrando un cráter, utilizando las siguientes ecuaciones.

$$X = X_0 + a \cos \theta \quad (2.3)$$

$$Y = Y_0 + b \sin \theta \quad (2.4)$$

En este trabajo se probaron las imágenes de CHANDRAYAAN-1 y del LRO, obteniendo buenos resultados a excepción de cráteres gemelos o aquellos que presentan el fenómeno de obliteración.

Así como se utiliza el ajuste de elipses siguiendo el error mínimo cuadrático, de igual manera en el ámbito del procesamiento de imágenes podemos encontrar trabajos [14] donde utilizan la técnica de la transformada de Hough para poder identificar formaciones circulares sean cráteres volcánicos o de impacto, y a partir de ellos hacer una clasificación. Comúnmente estos sistemas siguen el siguiente procedimiento:

### **Preprocesamiento de la imagen**

Lectura de las imágenes y selección de las zonas de interés, se realiza igualmente un resalte de ciertas características (bordes, contraste).

### **Detección de patrones**

Se obtienen las imágenes segmentadas utilizando algoritmos de segmentación multiespectral, o combinaciones de algoritmos similares.

### **Reconocimiento de patrones**

Lo patrones son encontrados, buscando formas circulares con la transformada de Hough.

Este método presenta buenos resultados, sin embargo hay que dar valores para los parámetros de segmentación y para la transformada de Hough y también engloba muchos elementos circulares que no son cráteres como lo son bordes montañosos curvos, depresiones, así como restos de volcanes.

Otra de las propuestas que encontramos en las que igualmente utilizan la transformada generalizada de Hough, en estas propuestas se apuesta por un método que contiene un sistema de segmentación de un “watershed” o cuenca, el cuál es útil para el registro de imágenes [40]. En [40] encontramos un detector automático de estructuras en el astro utilizando análisis para extraer características para registrar imágenes multitemporales.

El proceso empleado sigue el siguiente principio: Dado un par de imágenes para ser registradas, donde una es etiquetada como la entrada y la otra como la referencia, las características de ambas imágenes son extraídas para obtener la transformación óptima. La transformación óptima nos da los parámetros que son utilizados para registrar la imagen etiquetada como entrada con respecto a la de referencia, a detalle podemos describir el algoritmo de la siguiente manera:

1. Primero se realiza un preprocesamiento realizando un filtrado del ruido mediante un filtrado gaussiano y un filtro de mediana en cascada como observamos en [37].
2. Posteriormente para la detección de borde se utiliza un detector de Canny [4], como observamos en [18], logrando extraer los contornos en el gradiente de la imagen. Posteriormente, se le aplica el algoritmo de contorno “watershed” para obtener una imagen binarizada que muestre los contornos de la roca.
3. Se aplica un acumulador generalizado de Hough para identificar las distintas estructuras que nos ayudarán junto con los gradientes de oposición y con la ayuda de un acumulador en 3D, se estime el ángulo de dirección de la elipse y de esa manera se muestren los bordes de los cráteres.
4. Por último previo al registro, se aplica un acumulador estándar de Hough; propuesto en [8], para detectar las líneas rectas y de esa manera obtener las crestas de los cráteres.

Posterior a la etapa de detección, se prosigue con el método propuesto de registro en donde se busca una optimización del modelo de transformación

de una imagen que se propone el cual consiste en maximizar:

$$MOM(T) = \frac{1}{n} \sum_{(x,y); I(x,y) \neq 0} I_A(T(x,y)) \quad (2.5)$$

Donde  $MOM$  es la medición de igualdad,  $t$  es la transformación y  $x$  y  $y$  las coordenadas del plano.

A pesar de proponer un método para el registro de los cráteres, para la posterior detección de estos siguiendo la comparación de su “medición” de igualdad, una de las desventajas de este método es el factor de iluminación, el cual afecta el desempeño del algoritmo para la detección de los cráteres.

Entre estos métodos de igual manera encontramos algunas variantes enfocadas en el preprocesamiento para realizar una buena delimitación de los bordes o para una extracción óptima de la información. En [15] se utiliza un operador Canny adaptivo de acuerdo a las condiciones de la imagen, para ello se utiliza el Histograma de gradiente de magnitud la cual es efectivo para imágenes de cualquier tipo. El algoritmo para aplicar el filtrado GMH es el siguiente:

1. Se calcula el gradiente de magnitud  $g(i, j)$  y la dirección  $\alpha(i, j)$  de cada píxel utilizando el detector Sobel.
2. Se construye el GMH de cada píxel de una cresta obteniendo dos estadísticas: el gradiente de magnitud  $g_m$  que corresponde al número máximo de píxeles crestas, y el MSE  $\sigma_m$  el cual es calculado mediante:

$$\sigma_m = \sqrt{\frac{\sum_{i=1}^k h(g_i)(g_i - g_m)^2}{\sum_{i=1}^k h(g_i)}} \quad (2.6)$$

Donde  $k$  es el número de histogramas binarios y  $h(g)$  es la estadística de histograma del gradiente  $g_i \cdot g_m$ .

3. De ahí partimos para obtener el umbral de gradiente de magnitud  $J$  de los píxeles filtrados por el operador Canny, el cual se obtiene con la siguiente fórmula:

$$J_h = g_m + \sigma_m \quad (2.7)$$

4. Posteriormente el algoritmo realiza un pareado de los bordes como observamos en [18], uniendo de esa manera las regiones oscuras y claras en un borde circular-elíptico.

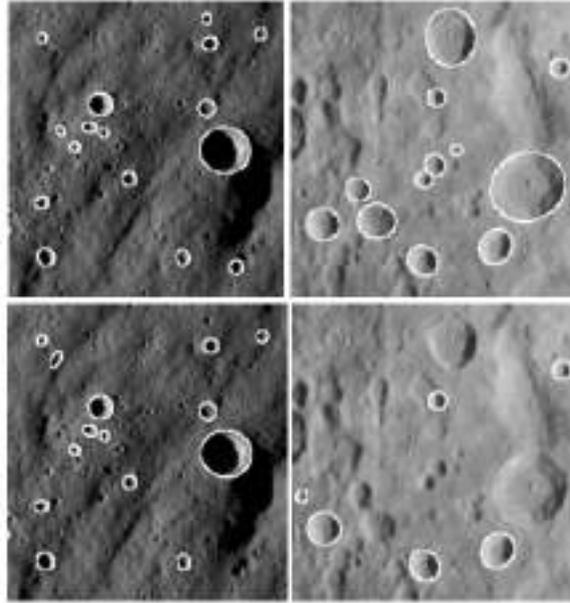


Figura 2.3: Comparación de los resultados obtenidos en [15]. Las dos imágenes de arriba representan el reconocimiento aplicando el *Canny adaptivo* y las inferiores con un *Canny normal*.

Esta propuesta [15] permite una mayor tasa de detección de muestras debido a la que el algoritmo adaptivo hace que la detección funcione en distintas iluminaciones y terrenos obteniendo una tasa de 85 % de reconocimiento con una tasa de falsas detecciones menor al 5 %.

Muchas propuestas de esta rama [22], [16],[33] siguen el mismo procedimiento: la primera fase en general consiste en una etapa de preprocesamiento de los datos, y a partir de ello realizar un ajuste de elipses o transformada de Hough. De igual manera observamos que se utilizan otras clases de datos para tener acceso a más información, o utilizan variantes morfológicas [36],[22], [3].

En esta rama igualmente encontramos proceso automáticos de detección que combinan varias propuestas para mejorar su tasa de desempeño como observamos en [35] en el cual se utilizan 4 propuestas para tener un clasificador compuesto. Las propuestas son las siguientes:

- Búsqueda de patrones de sombreado en la imagen.
- Búsqueda de características circulares en bordes de imágenes.
- Curvas y Círculos.
- Bordes circulares utilizando la transformada de Hough modificada para detectar circunferencias parciales.

Uniendo de esa manera los resultados de cada una de las propuestas, lo cual mejora el resultado pero sigue teniendo un porcentaje inferior a los métodos enfocados a herramientas de aprendizaje supervisado que estudiaremos a continuación.

## 2.2. Detección de Cráteres Lunares utilizando Métodos de Aprendizaje Supervisado

Detectar cráteres es una tarea difícil debido a que se presentan en distintas condiciones de luminosidad, tamaños, formas, etc. Como observamos en la anterior subsección, se han creado muchos métodos para realizar el conteo de manera automática, en base a la transformada de Hough, algunos utilizando algoritmos genéticos, morfología matemática, así como la combinación de múltiples técnicas [20, 35]. En particular decimos que existen varios aspectos que dificultan la elección adecuada de los parámetros para cada uno de los métodos distintos, entre ellos podemos encontrar los siguientes: El amplio rango de las dimensiones que puede tener los distintos cráteres que encontramos, así como los distintos niveles de conservación que tienen.

En [23] se propone el uso de un clasificador binario entrenado mediante el algoritmo de aprendizaje supervisado “*Adaptive Boosting*” el cual requiere

la utilización de características Haar, las cuales detallaremos en el siguiente capítulo. Como observamos en [42], este algoritmo busca obtener una función discriminante.

$$H(B) = \sum_{t=1}^T \alpha h_t(B) \quad (2.8)$$

Para de esa manera detectar las imágenes que sirvieron de entrenamiento. En [23] de igual manera se aplica una clasificación a distintas escalas, para posteriormente realizar una búsqueda del máximo local  $H(B)$ , este método propuesto alcanza una detección de cráteres iguales o mayores a 7 píxeles.

Una problemática encontrada en el anterior método es que se requiere tener una buena base de datos de imágenes representativa, es por eso que se buscan alternativas al “*Adaptive Boosting*” propuesto en [42] como en [25] en donde proponen representar los cráteres utilizando unidades C1, que corresponden a células complejas de la corteza visual. Este método sigue el proceso de reconocimiento de objetos en los seres vivos al ser detectados esos objetos por su corteza visual, para ello se generan características de la información extraída de las unidades C1, las cuales son invariables en tamaño y ubicación.

De cada mapa C1 podemos extraer un conjunto de características Haar mejoradas que contienen la información del gradiente de textura. Para ello las características del mapa C1 son embebidas en un espacio multidimensional, para posteriormente aplicar las características Haar y de esa manera tener una información más completa de las distintas texturas presentes. El algoritmo se detalla a continuación:

1. Se utiliza un arreglo multidimensional para representar la imagen.
2. Se obtiene la primera capa, un mapa S1. Haciendo una convolución de la imagen de entrada con un arreglo de filtros de Gabor con 4 orientaciones y 16 escalas.

$$F(x, y) = \exp\left(-\frac{x_0^2 + \sigma^2 y_0^2}{2\delta^2} x \cos\left(\frac{2\pi}{\lambda} x_0\right)\right) \quad (2.9)$$

En donde:

$$x_0 = x \cos(\theta) + y \sin(\theta) \quad (2.10)$$

$$x_0 = -x\sin(\theta) + y\cos(\theta) \quad (2.11)$$

- Posteriormente se crea la segunda capa, los mapas C1 a partir de las S1.

Utilizar la función de Gabor nos permite agrupar distintas orientaciones, escalas y rotaciones para englobarlas en 1 sola característica Haar, en el algoritmo propuesto se engloban 4 orientaciones y 16 escalas para obtener 64 filtros Gabor.

La preparación anterior permite encontrar un discriminativo intrínseco del sub-espacio para una clasificación precisa, para ello se utiliza un algoritmo de alineamiento local discriminativo que funciona minimizando la distancia dentro de las clases y maximizando la distancias entre clases para de esa manera obtener un modelo global.

En este trabajo [25] observamos que se obtiene una mejora significativa con respecto a [23] utilizando las plantillas Haar propuestas (C1) y generando las S1 con el filtro de Gabor.

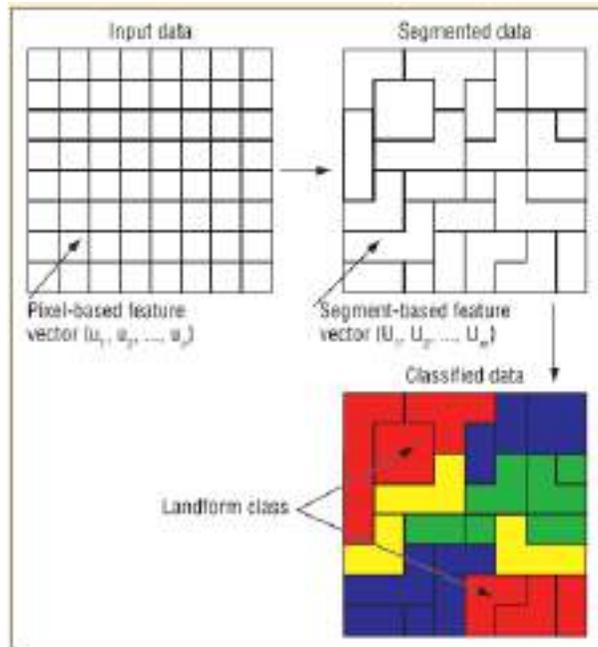


Figura 2.4: Creación de etiquetas de grupos de píxeles. (Tomado de [39])

Otra alternativa que encontramos es utilizar otras técnicas de aprendizaje supervisado o el uso de “*Naive Bayes*”. En [39] encontramos que se realiza un análisis de la imagen para dividirla de 64 píxeles a 23 segmentos, en donde los píxeles agrupados comparten similitud entre los elementos de su vector de características. En este caso se utiliza aprendizaje supervisado para poder etiquetar las distintas formaciones topográficas encontradas. Posteriormente se aplicó un algoritmo de SVM para entrenar el clasificador que podrá utilizar las clases para poder clasificar los segmentos que encontramos en una imagen ya procesada.

También encontramos otros trabajos donde se utilizaron “*SVM*” y “*Boosting*” con algunas modificaciones, en el caso de [43] donde realizan pruebas para la detección de cráteres marcianos. En este trabajo utilizan el mismo modelo de “*Adaptive Boosting*” utilizado en [42] sin embargo previamente realizan un entrenamiento neuronal previo a las 6 rondas de de “*Adaptive Boosting*” propuestas. Y en el experimento de “*SVM*” una vez obtenido el clasificador desarrollaron una implementación de FFT ( “*Fast Fourier Transform*” ) sobre la función de decisión, para de esa manera aumentar el funcionamiento en modelo de escalas muy superiores.

De igual manera podemos encontrar métodos basados en probabilidad volumétrica como en [2] en donde se utiliza una combinación de métodos para de esa manera descartar aquellas muestras que son filtradas previamente para formar parte de los “*Candidatos*”. En esta propuesta se sigue el siguiente proceso:

1. Para la etapa de detección de muestras candidatas realizamos dos pasos:

- Se realiza un recorrido con una máscara  $M$  a través de todos los píxeles para calcular el valor promedio local y observar si existe una gran desviación con respecto a los valores mínimos y máximos presentes en la máscara creando una nueva imagen definida por:

$$A_{u,v} = \max[m(M) - \min(M), \max(M) - m(M)] \quad (2.12)$$

- Posteriormente se obtiene el valor del umbral  $T$ , el cual ayuda a transformar la imagen para poder obtener solamente los bordes que contrastan de manera significativa con respecto a los demás.

2. Se realiza un *matching* de las muestras con una serie de templates calculados previamente con una implementación de FFT como se observa en [11]. Este *matching*, permite obtener la probabilidad que dado un píxel, este sea el centro de un cráter.
3. Por último la detección de los cráteres se realiza mediante un análisis probabilístico de los resultados del *matching*. De esa manera los píxeles que conforman la imagen realizan una comparación con el regional máximo del volumen probabilístico utilizando el siguiente criterio.

$$\forall x \in M \text{ para } P(x) = p. \quad (2.13)$$

$$\forall y \in \delta^{(1)}(M) / M \text{ para } P(y) < p. \quad (2.14)$$

Donde  $\delta^{(1)}$  es la dilatación morfológica de tamaño 1 y  $y$  es el punto del conjunto dilatado y a partir de ello se obtiene el valor regional máximo para detectar de esa manera los cráteres presentes.

$$R_p(P - 1) = \delta_p^{(i)}(P - 1) \quad (2.15)$$

Donde  $(i)$  es tal que  $\delta_p^{(i)}(P - 1) = \delta_p^{(i+1)}(P - 1)$ .

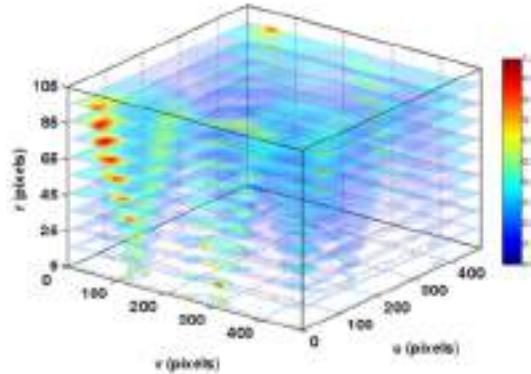


Figura 2.5: Volumen probabilístico obtenido de la FFT. (Tomado de [2])

En las propuestas anteriores observamos que se utilizan métodos de aprendizaje supervisado como lo son el “*SVM*” y el “*Adaptive Boosting*”, igualmente encontramos variantes de las características Haar [6] u otras características

las cuales son construidas mediante redes neuronales [23], así como otros patrones de construcción ya que nos permite obtener información de manera más eficaz como en [23, 7, 39, 43].

### 2.3. Discusión del Estado del Arte

En este mismo capítulo se presentaron algunos trabajos enfocados a la detección de cráteres en superficies de distintos astros. Varios de dichos trabajos utilizan distintos enfoques para abordar el mismo problema. En los trabajos, podemos distinguir que existen dos enfoques principales los cuales son el procesamiento de imágenes y el de aprendizaje supervisado. En el primero, la detección se realiza aplicando una transformación o extracción de información de una imagen, en cambio en el segundo se busca entrenar un clasificador capaz de detectar los cráteres mencionados.

Como podemos observar una vez identificado el cráter, se puede almacenar su ubicación para que posteriormente se realice la etapa del cálculo de los diámetros de estos, así como la distribución que presentan en distintas secciones de la superficie lunar. Detectar correctamente los cráteres con un mínimo margen de error es importante para obtener un cálculo muy fiable de la distribución de los cráteres lunares, eso implica el análisis de los distintos cráteres que podemos encontrar en la superficie. Debido a lo anterior es importante contar con una base de datos de imágenes, en este trabajo utilizamos una base de datos de imágenes que fue obtenida con la supervisión de un especialista con el conocimiento necesario para identificar cráteres presente en una imagen dada.

Dichos trabajos mencionados persiguen el objetivo de lograr la detección de los cráteres que se encuentran en una superficie dada, dicha detección nos sirve para posteriormente hacer un estudio de la distribución de ellos, o bien un estudio de la evolución topográfica de una sección en un lapso determinado de tiempo. En este trabajo se plantea abordar la detección de cráteres utilizando herramientas de aprendizaje automático, con base a una serie de características diseñadas para este problema en particular. Al igual que los trabajos enfocados al aprendizaje supervisado para detección de objetos, es necesario contar con un banco de imágenes que describiremos en la sección

de la metodología, para de esa manera poder entrenar el modelo en cuestión.

De igual manera se propone la utilización de un sistema de filtros para poder realizar un barrido de las muestras etiquetadas y rechazar los falsos positivos (cráteres detectados erróneamente), con el fin de tener una mejor tasa de detección. Lo anterior permitirá tener un buen detector de cráteres lunares lo cual es de suma importancia para trabajos futuros orientados en específico al cálculo de diámetros, circunferencias, así como el cálculo de distribución de cráteres sobre una superficie delimitada.

# Capítulo 3

## Marco Teórico

La observación astronómica es una actividad que ha permitido a lo largo de los años a los científicos estudiar fenómenos y astros que se encuentran en el sistema solar, y de igual manera fuera de él. Nos permite conocer, estudiar y entender lo que nos rodea. Es una actividad que ha tenido mayor auge, gracias al crecimiento de la tecnología la cual nos permite la utilización de herramientas tales como telescopios, satélites, entre otros para de esa manera estudiar los astros en cuestión, permitiendo resolver y explorar interrogantes que encontramos en nuestro sistema solar y fuera de él con mayor exactitud. Un ejemplo de los astros que encontramos en nuestro sistema solar es la luna; es el astro más cercano que gira alrededor de nosotros teniendo influencia en la gravedad terrestre, así como las mareas oceánicas, un astro que tenemos cerca y podemos ver a simple vista. La cercanía del astro ha permitido que desde el siglo XVI científicos como Galileo y Harriot utilizaran el telescopio para hacer los primeros mapas de la superficie lunar [29], de igual manera su cercanía permite que exista un interés en su estudio para conocer su historia y evolución [26]. A pesar de que los estudios y dibujos eran rudimentarios comparados con las imágenes de satélites actuales extraídas de satélites como el CHANDRAYAAN o como la misión de Apolo entre otros, la luna ha sido un objeto de estudio desde entonces.



Figura 3.1: Foto lunar fotografiada por Tony Spacey.

A simple vista se distinguen regiones más oscuras y otras más claras, sin embargo con la ayuda de un telescopio o binoculares podemos observar que una de las características de la luna que distinguimos en primera instancia es que se encuentra saturada de cráteres, los cuales tienen mayor presencia en algunas áreas y menores en otra. Con un estudio más detallado podemos observar que estos cráteres se presentan en distintos tamaños y formas. De igual manera podemos encontrar otras formaciones topográficas como lo son valles, canales entre otros [29], las cuales han sido motivo de varios estudios topográficos.



Figura 3.2: Cráteres presente en la superficie lunar. Imagen obtenida de la misión Apollo [1].

### 3.1. Cráteres Lunares

Los cráteres espaciales son estructuras formadas por un proyectil cósmico lo suficientemente grande para penetrar una atmósfera de un astro con poca desaceleración o ninguna, y que colisiona en la superficie de este a velocidades mayores a  $11\text{Km/s}$  [10]. Los impactos por estos proyectiles al ocurrir a una velocidad mayor que la velocidad que el sonido tiene en la superficie impactada, esto ocasiona que exista una deformación en el momento de la colisión. Dicha deformación ocurre debido a que se forman ondas de choque intensas que generan una gran presión que no es producida por procesos geológicos ordinarios, y al expandirse producen el hundimiento y modificación de la superficie. En la luna, los cráteres formados por el impacto de objetos tienen un diámetro aproximado de 10 veces el diámetro del objeto que lo impactó.

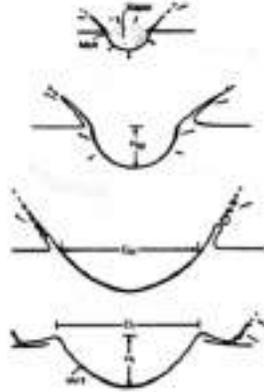


Figura 3.3: Proceso de formación de un cráter, se impacta el proyectil y posteriormente la onda de choque deforma los bordes y expande el diámetro del cráter [24].

Los impactos de cráteres en superficies planetarias proveen información importante sobre la edad, naturaleza y composición de la superficie en donde están presentes. La densidad de los cráteres está estrechamente relacionada con la edad de la superficie y su estimación es utilizada para establecer la evolución de los terrenos [24]. Es por ello que una vez con la información de la densidad de cráteres, es posible encontrar una evidencia de algún evento o fenómeno ocurrido en la superficie como lo pueden ser la actividad sísmica, flujo de lava, formación de cráteres volcánicos, así como la erosión y deformación de la superficie planetaria producida por el impacto de objetos a altas velocidades.

### 3.2. Estimando la edad de la superficie

Se ha comprobado que los planetas como Mercurio, Venus, Marte y la luna muestran similitudes en la características de la distribución que nos dicen que existe una familia de cuerpos en el sistema que son responsables de los cráteres presentes en los planetas del sistema solar[27, 28]. Esto es de gran relevancia debido a que se puede estimar la edad de una superficie de los astros cuyas rocas no han sido recopiladas y medidas gracias una técnica

desarrollada en los 1960's que consiste en el conteo de cráteres.

En el caso de la luna sabemos que los cráteres de gran tamaño como el *Imbrium* fueron rellenados por lava, formando los actuales mares, abarcando dichos mares grandes extensiones de la superficie lunar cubriendo hasta un 18% de la superficie de la luna y la mayoría de la cara visible (el 38%) [32], y las zonas no cubiertas son anteriores a la formación de los mares y por ende son más antiguas. A través del paso de los años, la frecuencia de los impactos fue reduciéndose gradualmente, y eso explica porque la frecuencia de los impactos en los mares es menos frecuente, siendo superficies más jóvenes. La Luna al no presentar procesos corrosivos, los cráteres se mantienen permitiendo que se establezcan conteos para calcular la edad de la superficie donde las tierras de la luna pueden tener antigüedades de más de 3.8 mma y los mares entre 3.2 mma y 3.8 mma.

De lo anterior podemos deducir que el número de cráteres presentes en una superficie planetaria está determinado por la edad de la superficie y el promedio de la tasa de producción de cráteres, permitiendo que al conocer uno de los datos de un aspecto, pueda estimar el otro. Los mares de la luna han preservado cierto número de cráteres a lo largo de sus 3.2 a 3.5 Ga de historia, esas superficies calibradas permiten el conteo de cráteres como herramienta para poder calcular la edad de otras superficies de la luna. De igual manera al obtener una gráfica de frecuencia con tamaño de los cráteres observamos que la distribución de la luna al compararla con planetas podemos obtener cierta similitud de la distribución medida. En el trabajo de Ivanov [17], se observa que presentan la misma forma de distribución tamaño-frecuencia que presentan los asteroides del cinturón de asteroides.

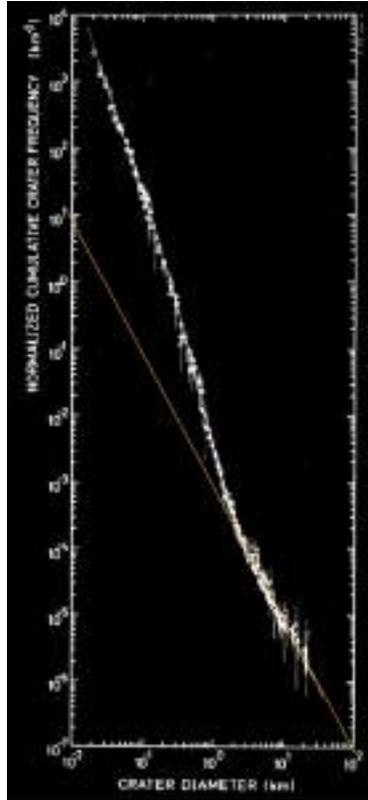


Figura 3.4: Gráfica de distribución frecuencia-tamaño. Neukum et al [27]

La forma que presenta la curva de la distribución de pequeños cráteres primarios encontrados en la luna, Marte, Mercurio es muy parecida, aunque al comparar la distribución de los cráteres presentes en la luna y los terrestres hay una discrepancia, utilizando un posterior análisis se confirma que la producción de la distribución tamaño -frecuencia es parcialmente incorrecta debido a que un número grande de cráteres que se incluyen son más antiguos que la era Erastotenia, para ello al ajustar las curva de distribución ahora si observan resultados similares, apoyándose en la afirmación Neukum [27] que existe una familia de objetos que son responsables de la formación de la mayoría de los cráteres presentes en el sistema solar. Para ello se realiza una gráfica de distribución de frecuencia-tamaño; la cual nos permitirá estudiar según lo comentado anteriormente 3 aspectos importantes:

- La edad de la superficie en cuestión en comparación con la edad de otras superficies cuyas gráficas de distribución frecuencia-tamaño disponemos.
- Claves de la evolución de la superficie y del astro en cuestión.
- De igual manera será una clave para estudiar las superficies en las que se encuentran.

### 3.3. Procesamiento de Imágenes

La detección de objetos en una imagen puede ser abordada desde distintas perspectivas entre las cuales encontramos la perspectiva de la morfología del objeto, así como los recursos disponibles para la detección de esos. Lo anterior nos lleva a dos caminos para la detección de esos: el procesamiento de imágenes y las herramientas que nos dan el aprendizaje automático. Incluso, en ocasiones ambas ramas se complementan, ya que mayormente el aprendizaje máquina precisa de información de una imagen cuando esta ya sufrió alguna transformación de su información haciendo más relevante ciertos datos.

La clasificación y detección de imágenes contenidas en una superficie requiere la manipulación de la información contenida en una imagen para extraer de ciertos grupos de píxeles. La información que utilizaremos para detección de estas comparándolas con un clasificador  $h(x)$  el cuál define las características que hacen pertenecer dicho grupo de imágenes a cierta clase de objeto.

De acuerdo con [12], el procesamiento de imágenes digitales se refiere a los procesos que reciben y devuelven una imagen incluyendo a los procesos que extraen atributos de las mismas.

Para el procesamiento de imágenes es necesario tener una representación del objeto, dicha representación es obtenida del entorno y posteriormente guardada por una matriz de valores de  $m$  filas por  $n$  columnas:

$$B_{m-1,n-1} = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,n-1} \end{pmatrix} \quad (3.1)$$

La cual es obtenida al discretizar la imagen con algún software que lea las imágenes dadas. De esa manera con la representación dada por la matrix B, tenemos una imagen representada por una función de dos variables  $I(x,y)$ ; en la cual se puede hacer referencia a un píxel dentro de la imagen  $(x,y)$ . Esta función tiene un dominio relacionada con la dimensión de la imagen donde  $0 \leq x < ny0 \leq y < m$ . Los valores generalmente pueden ser de varios canales pero usualmente se utiliza un solo canal de grises donde se almacenan valores entre 0 y 255 según los valores de intensidad, donde el cero representa ausencia de iluminación (negro) y el 255 presencia de iluminación al máximo (blanco) [12]. Una vez representada la imagen, podemos acceder a los valores de los píxeles llamando la función  $I(x,y)$ , así como los píxeles vecinos que rodean a este  $I(x-1,y), I(x,y-1), I(x+1,y), I(x,y+1)$ , de esa manera los píxeles que se encuentran en cierto rango de un punto central, formando subventanas a las cuales se les extraería la información para su posterior clasificación.

### 3.3.1. Imagen Integral

Muchos métodos de clasificación requieren la extracción de información tanto en las imágenes de entrenamiento como en las imágenes de prueba, para ello existen diversas características que se pueden apreciar en una región como lo son su derivada, dirección de la luz, gradientes orientados de histogramas, características tipo Haar, entre otras. Es importante utilizar herramientas que permitan obtener la sumatoria de regiones de la imagen, ya que al realizar una sumatoria cada vez que se requiera incrementa el costo del cálculo global. Una herramienta que soluciona esta problemática es la obtención de una representación de la imagen intermedia, la cual calcula de una sola vez la sumatoria de las imágenes, y a la cual se pueda acceder de manera rápida. La representación utilizada es llamada imagen integral y fue propuesta por Viola y Jones [42].

La imagen integral en un punto  $(x, y)$  de la matriz contiene la sumatoria de todos píxeles contenidos en la región rectangular formada por el origen de la imagen en  $(0,0)$ , hasta un punto dado como observamos en la siguiente figura:

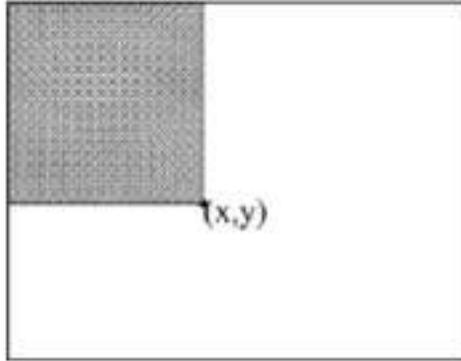


Figura 3.5: Rectángulo que representa la imagen integral en el punto  $(x,y)$ .

La imagen integral en una ubicación dada  $ii(x, y)$  la podemos definir como:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3.2)$$

Donde  $i(x, y)$  es la intensidad del píxel  $(x, y)$ . El cálculo de la imagen integral, se hace en una sola vez, guardando los valores de cada punto en un arreglo bidimensional. Pasando una sola vez por todos los píxeles  $(x, y)$  de la imagen, de esa manera para calcular el valor de la característica basta con hacer 3 operaciones aritméticas de esta manera se ahorra un tiempo considerable, en lugar de estar calculando la sumatoria cada vez ya que solo se necesita acceder a 4 puntos independientemente del tamaño de la imagen.

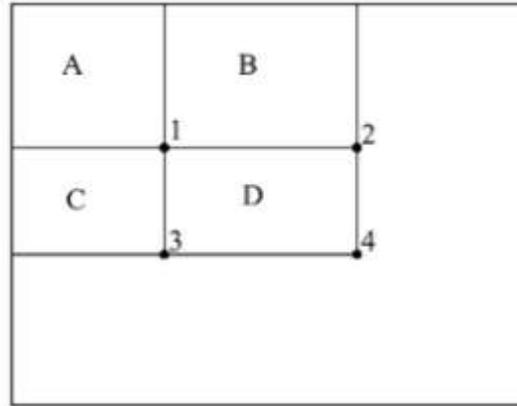


Figura 3.6: El valor de la imagen integral en la ubicación 1 es la suma de los píxeles comprendidos en el rectángulo A, es decir  $ii(1)=A$ . Análogamente,  $ii(2)=A+B$ ,  $ii(3)=A+C$ ,  $ii(4)=A+B+C+D$ . De esto se deduce que la suma de los píxeles comprendidos en D es  $ii(4)+ii(1)-[ii(2)+ii(3)]$ , para lo cual se requieren las 3 operaciones aritméticas mencionadas (dos sumas y una resta).

### 3.3.2. Plantillas de Convolución

Existen otras herramientas importantes para la extracción de características como lo son las plantillas de convolución, las cuales se pueden representar como una matriz de  $m \times n$ , donde se le asigna a cada elemento de la matriz la intensidad del píxel correspondiente. Análogamente en imágenes RGB la matriz se vuelve de  $m \times n \times 3$  donde cada entrada de la matriz  $(x, y, z)$  representa la intensidad del píxel en la posición  $(i, j)$  en el canal  $k$ .

En el proceso de extracción de características es necesaria la extracción de valores de grupos de píxeles, lo cual es realizada al aplicarlas una plantilla de convolución. La plantilla de convolución se encargan de obtener un nuevo valor del píxel en la posición  $(i, j)$  aplicando dicha plantilla como observamos en la siguiente figura, en donde los vecinos juegan un papel importante para el nuevo cálculo del píxel. Estas plantillas pueden variar en cuanto tamaño y configuración, y se adaptan a la morfología que queramos utilizar.

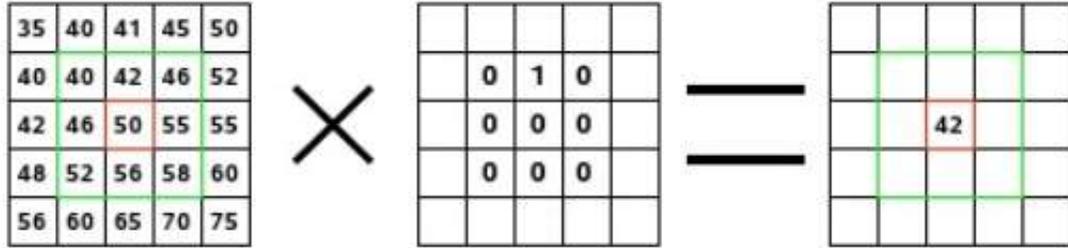


Figura 3.7: Ejemplo de aplicación de una plantilla de convolución.

Existen diversas plantillas que pueden representar desde filtros y derivadas de una imagen, así como las características tipo Haar que observaremos en la siguiente sección.

Las plantillas que se encargan del filtrado de las imagen consisten en la multiplicación de una máscara o kernel como mencionamos previamente. La fórmula para aplicar el filtrado está dada por:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b A(s, t)I(x + s, y + t) \quad (3.3)$$

Donde  $A$  es la máscara de convolución,  $I$  la imagen a filtrar.

La operación del filtrado de imágenes, de igual manera puede ser aplicada para la el cálculo de la derivadas de una imagen utilizando los siguientes kernels:

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Donde la primera matriz representa la derivada con respecto a  $x$  y la segunda la derivada con respecto a  $y$ .

### 3.3.3. Plantillas Haar

El concepto de la característica de tipo Haar fue introducido en 1997 por Papageorgiou et. al. [30]; donde se buscaba la representación de la imagen que contuviera los valores de las intensidades promedio de ciertas regiones, para de esa manera tener una característica adecuada para la clasificación y segmentación de la imagen. Las primeras características consistieron en 4 distintas configuraciones de rectángulos las cuales en el trabajo de Papageorgiou se enfocaban en la detección de peatones en entornos abiertos, dichas configuraciones consisten en las siguientes:



Figura 3.8: Plantillas Haar: una de configuración vertical, otra horizontal, una diagonal y la última de 3 rectángulos.

Estas características fueron introducidas para la detección de objetos, y más específicamente de rostros en el trabajo de Viola y Jones [42], donde utilizarían este modelo para utilizar su clasificador binario entrenado por AdaBoost.

Estas plantillas se aplican sobre las muestras sobre algún píxel de la imagen, los valores de las plantillas solo tienen valores de 1 o -1 según el color (negro o blanco), lo cual lo define como una plantilla de convolución binaria. Al aplicar las plantillas obtenemos un número que proviene de la sumatoria de los valores de los píxeles dentro del rectángulo blanco menos la sumatoria de los valores de los píxeles que conforman el rectángulo negro.

## 3.4. Aprendizaje Automático

El aprendizaje automático es una rama de la inteligencia computacional que se encarga del estudio de los algoritmos que permiten a un programa

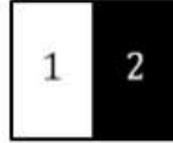


Figura 3.9: Plantilla de convolución. Los valores de intensidad son restados según la configuración donde el negro representa el -1 y blanco 1.

“aprende”. Un aprendizaje se logra cuando es capaz de modelar un objeto o comportamiento que tiene que reconocer, o de igual manera cuando se cumplen ciertas condiciones que satisfacen el modelo obtenido. El modelo obtenido debe aprender las características que reconocerá, para ello debe agarrar experiencia reconociendo un objeto o suceso dada una serie de características, y en base a ello aprender los parámetros que definimos como el error (muestras detectadas erróneamente). El aprendizaje automático es muy utilizado para el reconocimiento de rostros para identificación y registro de personal de una empresa, reconocimiento de huellas dactilares, reconocimiento de iris, entre otros [42],[31], [34],[19].

Una de las características de esta rama del aprendizaje automático es que se requiere realizar un proceso de retroalimentación del algoritmo para aprender de los errores, y de esa manera en la siguiente iteración de aprendizaje el algoritmo elige mejores parámetros para tener un error menor. Esta retroalimentación nos permite dividir el aprendizaje automático en tres ramas principales [21]:

- **Aprendizaje supervisado:**

En este tipo de aprendizaje se requiere de un conjunto de entrenamiento el cual está conformado por entradas de distintas clases, la cual la función analizará, junto con la salida (el error) y dependiendo de este realizar un ajuste para lograr que la salida sea lo más óptima posible.

- **Aprendizaje No supervisado:**

En este tipo de aprendizaje el modelo utiliza el mismo conjunto de entrenamiento, pero desconoce cuál es la salida esperada, sin tener esa comparación que observamos en el aprendizaje supervisado.

- **Aprendizaje por refuerzo:**

Por último, tenemos el aprendizaje por refuerzo, el cual consiste en realizar estímulos al entorno que rodea al agente inteligente y este va aprendiendo en base a la recompensa que le genera alcanzar dichos estímulos; existiendo estímulos que entregan una mejor recompensa que otros.

### 3.4.1. Clasificación Binaria

Los métodos mencionados previamente, requieren un modelo que represente numéricamente los elementos que interactúan con el problema, haciendo que la información requerida pueda ser agrupada en vectores que contenga un valor específico para cada información necesaria para establecer la predicción. Este conjunto de valores son llamadas características, las cuales conforman la entrada; tanto para el entrenamiento del modelo, como para la predicción de un resultado. Comúnmente se obtienen al aplicar plantillas de convolución [5],[42], gradientes [12], así como filtrados de la imagen que modifican esta para resaltar las características mencionadas, de igual manera es posible emplear directamente el valor de los píxeles como características. Al proceso de selección y extracción de la información se le conoce como extracción de características [13]. Posteriormente dichas características extraídas, juegan un papel fundamental en el entrenamiento de los modelos, llegando a la etapa de la clasificación donde el modelo asigna correctamente un conjunto de clases y/o etiquetas a un conjunto de objetos que conforman la prueba, podemos resumirlo como la separación de un objeto con los no objetos, dando lugar a la identificación y detección de estos.

En este trabajo se empleó una clasificación binaria donde el conjunto de clases posibles del total solo son dos, siendo posible asignar dos etiquetas según la salida del modelo; la primera es el objeto en cuestión y la otra es cuando el grupo de píxeles no representan el objeto (ausencia del objeto).

En este problema interviene un conjunto de instancias  $C$  pertenecientes a un conjunto mayor en donde existe un conjunto de etiquetas  $y$  con dos posibles etiquetas  $+1,-1$  o en algunos casos  $+1,0$ , en donde a cada instancia se le asigna una etiqueta. La función del clasificador binario consiste en asignar

la etiqueta que representa la clase a la cual pertenece la instancia/objeto a evaluar.

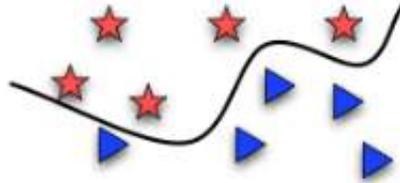


Figura 3.10: Ejemplo de Clasificador binario. En esta imagen observamos que se dividen las instancias en dos clases (estrellas y triángulos, el modelo que clasifica las muestras es la ecuación de la línea curva dando lugar a que las muestras encima del objeto sean estrellas y las de abajo triángulos.

El método de AdaBoost será descrito a continuación, al igual que los SVM (*Support Vector Machines*), los cuales se emplearon para la detección de grupos de píxeles con características que representen a un cráter dentro de una región de la superficie lunar.

### 3.4.2. Adaboost

El algoritmo de Adaboost es un modelo de clasificación utilizado para la detección de objetos con base a la creación de un clasificador fuerte a partir de otros clasificadores llamados débiles. En un problema de clasificación binaria donde tenemos un conjunto de objetos a clasificarse, un clasificador débil es aquel que es ligeramente mejor que una clasificación azarosa igualmente ponderada, este clasificador por lo general tiene un alto porcentaje de error. El algoritmo de Adaboost detecta cual es el clasificador  $h_n$  con el menor error posible que divida toda las instancias y pondera las instancias con un nuevo peso para que el siguiente clasificador  $h_n + 1$  se enfoque en las muestras detectadas erróneamente. Este proceso se repite iterativamente tantos clasificadores débiles  $h_1, h_2, h_3 \dots h_m$  conformen el clasificador final o fuerte, el algoritmo utilizado en [23, 42] es explicado a detalle a continuación:

1. Se asignan pesos iniciales de las instancias/objetos,  $D_1(i) = 1/m$  (Donde  $i$  es la instancia/objeto en cuestión y  $m$  el número total de instancias/objetos).
2. El algoritmo de AdaBoost elige el siguiente clasificador débil, para ello se hace un barrido de los distintos clasificadores posibles, y se obtiene el error de cada uno de ellos con la siguiente fórmula:

$$error_t = \sum_{i=1}^m D_t(i)[y_i \neq h_j(x_i)] \quad (3.4)$$

El clasificador débil  $h_t$  se elige seleccionado aquel que tenga el menor error con los pesos  $D_t$  actuales.

3. Posteriormente para la actualización de los nuevos pesos  $D_{t+1}$  y el cálculo que tiene el clasificador débil, se calcula el  $\alpha_t$ , el cual está dado por:

$$\alpha_t = \frac{1}{2} \log \left( \frac{1 - error_t}{error_t} \right) \quad (3.5)$$

4. Con el  $\alpha_t$  obtenido se realiza la actualización de los pesos que nos servirán para elegir el siguiente clasificador débil  $h_t$ , dicha actualización la realizamos mediante la siguiente fórmula:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad (3.6)$$

Donde  $Z_t$  es un factor de normalización para hacer que  $D_{t+1}$  sea una distribución, y se obtiene mediante la siguiente fórmula:

$$Z_t = \sum_{i=1}^m h_t \quad (3.7)$$

El proceso se repite  $N$  número de veces hasta tener  $N$  clasificadores débiles con sus respectivos pesos:

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right) \quad (3.8)$$

Siendo  $H(x)$  el modelo entrenado,  $h_t$  el valor que le otorga el clasificador débil,  $t$  al descriptor y  $\alpha$  el peso que tiene ese clasificador débil. La sumatoria final, nos muestra la clasificación correspondiente de los descriptores, este tendrá un valor mayor a cero si es el objeto detectado, y menor o igual a él si no lo es.

Para ilustrar de manera más sencilla el algoritmo utilizaremos la figura 3.11, donde se puede observar un conjunto de puntos (rojos y azules) que conforman las muestras a clasificar. Dichas muestras al estar en un plano de dos dimensiones deben ser separados por líneas en este caso son líneas con distintas pendientes, las cuales separan un grupo de muestras de otras. El algoritmo es descrito a continuación en concordancia con la imagen.

1. En la primera iteración observamos que se separan algunas muestras con una recta encima del cúmulo de descriptores de color azul, como hacen falta separar más puntos rojos pasamos a la iteración dos, no sin antes obtener un peso de acuerdo al error obtenido.
2. En la segunda iteración el algoritmo se enfoca en las muestras clasificadas erróneamente, generando un nuevo clasificador débil (recta con distinta pendiente), asignando un peso al nuevo clasificador.
3. El proceso se repite  $n$  veces donde  $n$  es el número de clasificadores que se requieran encontrar.
4. Al final obtenemos una sumatoria de todos los clasificadores donde cada uno tiene un peso distinto y conforman el clasificador fuerte.

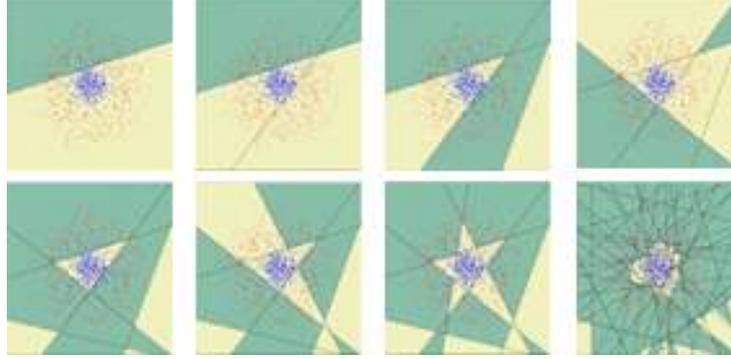


Figura 3.11: Ejemplo de la iteración para la elección de los clasificadores débiles en las iteraciones 1,2,3,4,5,6,7 y 8 que es el clasificador final o clasificador fuerte.

### 3.4.3. SVM (*Support Vectors Machines*)

Los SVM conforman un algoritmo de aprendizaje supervisado que se encarga de entrenar un modelo de clasificación a partir de un conjunto de entrenamiento. Al igual que el AdaBoost este algoritmo se encarga de asignar la clase correcta a futuras instancias/objetos. Con los SVM, la clasificación se realiza de acuerdo a la siguiente expresión:

$$h_{w,b}(x_t) = f(w^T x_t + b) \quad (3.9)$$

Donde  $h_{w,b}(x_t)$ , es el clasificador que se encarga de asignar etiquetas,  $w$  es el vector de pesos,  $b$  el bias del hiperplano y  $x_i$  es un vector descriptor con las características que conforman el objeto a clasificar.  $f(z)$  conforma la función que etiqueta los objetos en donde asigna el valor de 1 cuando los valores de  $z$  son mayores a cero, y asigna la etiqueta de -1 en valores iguales o menores a cero. Para lograr esto las SVM buscan poner solución al siguiente problema de optimización:

$$\begin{aligned} \min_{w,b,\xi} &= \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \\ \text{s.t.} & \gamma_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i > 0 \end{aligned} \quad (3.10)$$

Donde  $C$  es el parámetro de penalización.

Para realizar el análisis de patrones, se utilizan kernels que se encargan de mapear los datos a otro espacio dimensional que nos permitirá encontrar de manera más sencilla el plano o hiperplano que nos ayudará a separar las muestras en las distintas clases. La función kernel  $K$  se encuentra definida como:

$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j) \quad (3.11)$$

Usualmente se utilizan 3 kernels básicos: el lineal, el polinomial y la función de base radial:

$$K(x_i, x_j) = x_i^T x_j \quad (3.12)$$

$$K(x_i, x_j) = (px_i^T x_j + r), p > 0 \quad (3.13)$$

$$K(x_i, x_j) = \exp(p\|x_i - x_j\|^2), p > 0 \quad (3.14)$$

# Capítulo 4

## Metodología

En este capítulo, se presenta el procedimiento empleado para la detección de los cráteres lunares presentes en una imagen de alguna sección de la superficie lunar. Como mencionamos anteriormente, el procedimiento empleado de extracción de características con las máscaras propuestas obtiene alrededor de 10,000 distintas características tipo Haar. Posteriormente, con las características obtenidas se entrenó un modelo clasificador con el algoritmo de “Adaptive Boosting”. Por último con el modelo obtenido se implementó un sistema de detección de cráteres a partir de filtros para separar los cráteres obtenidos de las muestras negativas. La obtención de dichas características empleadas serán explicadas a mayor detalle a continuación, así como el algoritmo de “Adaptive Boosting” utilizado, de igual manera se explicará el filtrado de muestras positivas/negativas realizado en el clasificador final.

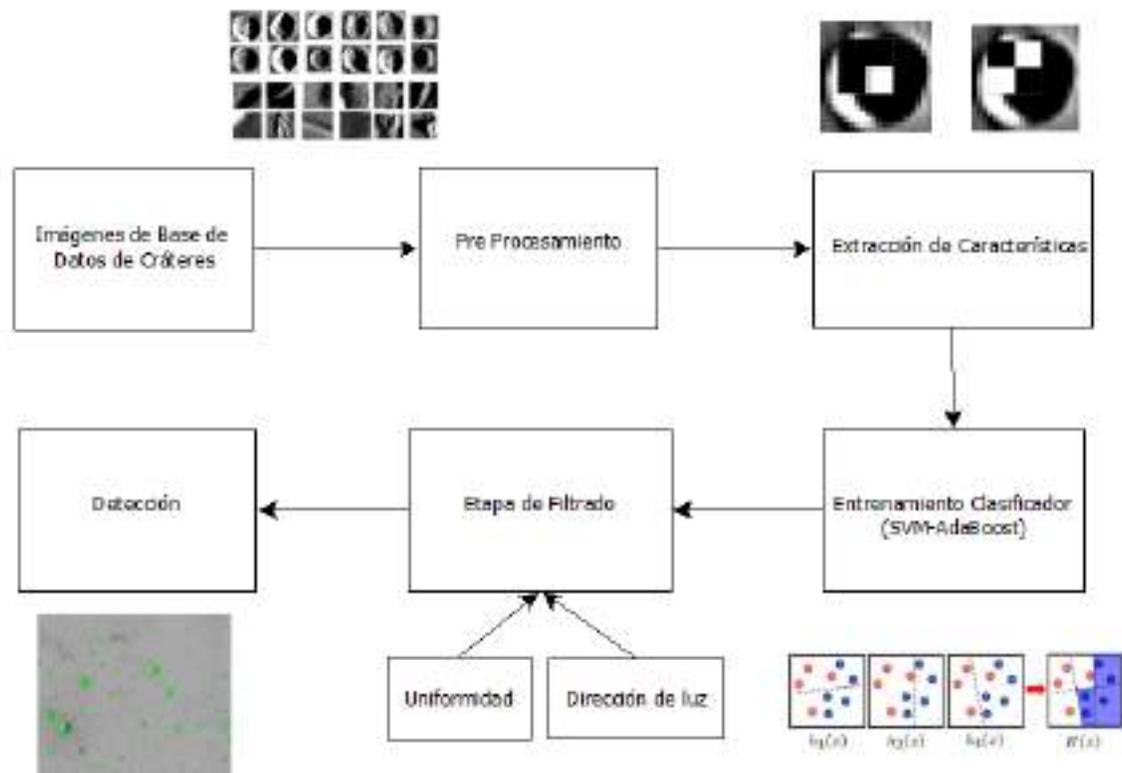


Figura 4.1: Proceso general empleado para la detección de cráteres.

## 4.1. Preparación de las Muestras de Entrenamiento

Para el entrenamiento del modelo se realizaron recortes de imágenes obtenidas del banco de imágenes de la misión Apollo. Las imágenes fueron extraídas de distintas imágenes fuente que conformaban superficies con cráteres, así como zonas con poca distribución de cráteres. De igual manera se utilizaron imágenes que contenían otras formaciones topográficas que podemos encontrar en la superficie lunar. A continuación podemos observar 2 de las imágenes de donde se extrajeron los cráteres que conformaron nuestro con-

junto de entrenamiento y de prueba, así como las imágenes de los objetos que no son cráteres:

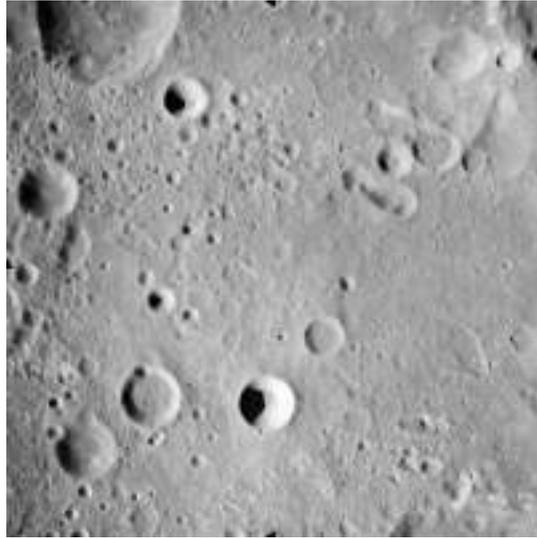


Figura 4.2: Imagen fuente 1.



Figura 4.3: Imagen fuente 2.

A partir de las cuales se realizó la extracción de las muestras positivas y negativas que conforman nuestro banco de datos, se cuidó que la relación del corte de los cráteres (muestras positivas) sea menor a 2.75 veces el radio del cráter y que sean recortes de 24x24 píxeles, de igual manera los recortes se hicieron con el cráter centrado, para posteriormente rotarlas 23 veces, y de esa manera tener 24 cráteres distintos de uno solo obteniendo cráteres similares pero con distinta orientación de la dirección de la luz. Las rotaciones fueron hechas rotando el cráter original 15 grados a la vez.

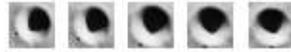


Figura 4.4: Cráter original con 4 rotaciones.

A partir de lo anterior se obtuvo una base de datos de 960 muestras positivas. Para el conjunto de muestras negativas, se realizaron 240 recortes de formaciones topográficas de superficies que no contenían o representaban cráteres, y posteriormente se hicieron rotaciones de  $90^\circ$  para de esa manera obtener las 960 muestras negativas.

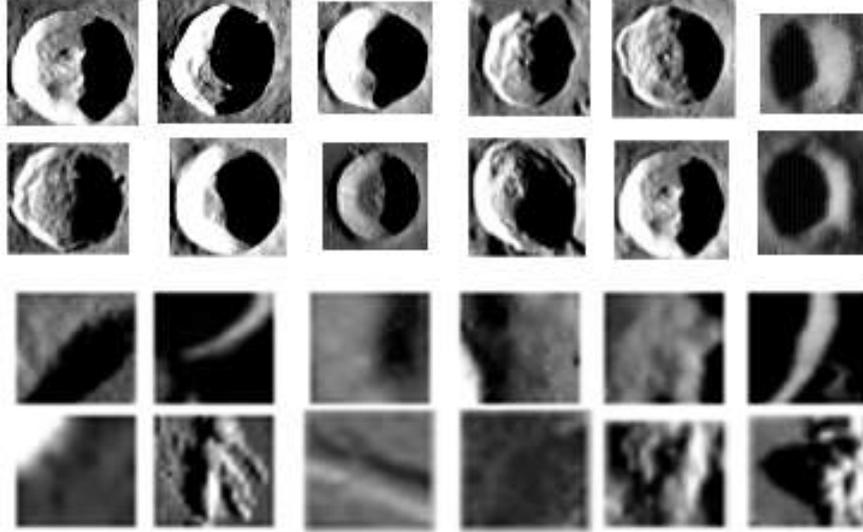


Figura 4.5: Algunos ejemplos de las muestras que se utilizaron para el entrenamiento.

## 4.2. Preprocesamiento de las Muestras

Posteriormente se prepararon las muestras para de esa manera tener un conjunto de muestras generalizadas, ya que el mismo proceso de preprocesamiento se le aplicó de manera posterior a las imágenes que sirvieron como pruebas.

Primero, se transformaron las imágenes a un solo canal de color (grises) en caso que algún elemento de la muestra poseyera 3 canales de color (RGB) siguiendo la siguiente fórmula:

$$x = 0,299r + 0,587g + 0,114b \quad (4.1)$$

De esa manera obtenemos un valor de luminosidad  $x$ , a partir de los valores de Rojo ( $r$ ), Verde ( $g$ ) y Azul ( $b$ ).

Posteriormente, se aplicó una ecualización del histograma de la imagen  $K$  para mediante una transformación obtener un histograma con una distribución uniforme de la imagen, logrando que de esa manera exista el mismo número de píxeles para cada nivel de gris en una imagen en tonos grises, para ello se aplicó la siguiente función:

$$v(i, j) = \frac{Fu(u(i, j) - Fu(a))}{1 - Fu(a)}(L - 1) + 0,5 \quad (4.2)$$

Logrando maximizar el contraste de la imagen  $K$ .

### 4.3. Extracción de Características

En todo algoritmo para entrenar un clasificador, las características juegan un papel muy importante, una buena elección de características nos permite agrupar y seleccionar información (píxeles) que describan el objeto de estudio para así entrenar correctamente el clasificador. Para la extracción de características utilizamos las plantillas Haar en donde extraemos poco menos de 10,000 características de cada imagen escalada de 24 x 24 píxeles.

En primera instancia, la imagen que contiene el cráter centrado o la formación topográfica que no es cráter (muestra negativa), es transformada en escala en grises, posteriormente escalada desde su tamaño original a un tamaño fijo de 24x24 píxeles, posteriormente se realiza el siguiente algoritmo que detallaremos posteriormente para la extracción de las características utilizando distintas configuraciones de plantillas Haar como observamos en el capítulo del Marco Teórico, las plantillas nos ayudaron a extraer la información presente en las imágenes que conforman nuestro banco de imágenes.

Antes de explicar el algoritmo que realizamos para la utilización de las plantillas, explicaremos el método realizado para la división de una imagen en 4 secciones o 9 secciones diferentes según sea el caso para de esa manera asignar una etiqueta a cada una de las secciones, en donde se asigna un valor que es el resultado de la sumatoria de los píxeles que conforman dicha región, para posteriormente sumarse o restarse a las demás regiones según sea la configuración a aplicarse.

Una vez establecido el ancho y el largo de la subventana, así como el punto inicial de su trazado dentro de la imagen, se realiza el siguiente procedimiento para establecer las regiones:

Para establecer 4 regiones, se hace lo siguiente:

- Se establecen los puntos iniciales  $X_1$  y  $Y_1$ .
- El ancho con respecto al eje X es utilizado para sacar los puntos  $X_2$  y  $X_3$ , para obtener de igual manera el extremo derecho, y el punto medio ( $\text{anchoX}/2$ ) con respecto al eje X.
- El ancho con respecto al eje Y es utilizado para sacar los puntos  $Y_2$  y  $Y_3$ , para obtener también el extremo derecho, y el punto medio ( $\text{anchoY}/2$ ) con respecto al eje Y.

Lo cual nos arroja una cuadrícula que representa la subventana, esta es dividida en 4 secciones diferentes pudiendo adoptar configuraciones como las que observamos en la siguiente figura:

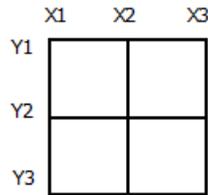


Figura 4.6: Coordenadas de la cuadrilla de 2x2.

Para 9 regiones, se realizó un proceso similar:

- Se establecen los puntos iniciales  $X_4$  y  $Y_4$ .
- Aquí el ancho con respecto al eje X se divide en 3 partes, en donde:
  1.  $X_5 =$  coordenada del punto inicial  $X_4 + \text{anchoX}/3$ .

2.  $X_6 =$  coordenada del punto inicial  $X_4 + \text{anchoX} * (2/3)$ .
  3.  $X_7 =$  coordenada del punto inicial  $X_4 + \text{anchoX}$ .
- De igual manera para obtener las coordenadas con respecto al eje Y se requiere dividir el ancho 'Y' en 3 partes:
    1.  $Y_5 =$  coordenada del punto inicial  $Y_4 + \text{anchoY}/3$ .
    2.  $Y_6 =$  coordenada del punto inicial  $Y_4 + \text{anchoY} * (2/3)$ .
    3.  $Y_7 =$  coordenada del punto inicial  $Y_4 + \text{anchoY}$ .

Las cuales nos dan herramientas para dividir la subventana en 9 regiones distintas, permitiendo que se utilicen las configuraciones deseadas, aumentando el número de posibilidades para una mayor cantidad de plantillas diferentes para la extracción de características.

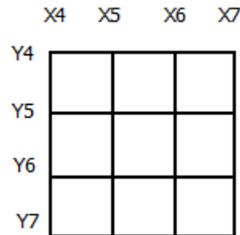


Figura 4.7: Coordenadas de la cuadrilla de 3x3.

Lo anterior nos detalla cómo se establecen las coordenadas, para realizar las sumatorias de las distintas regiones, y a partir de la configuración en cuestión, sumar o restar los resultados de cada una de las regiones, siendo las sumatorias de las regiones representadas por el color negro restadas aritméticamente a las regiones representadas por el color blanco; dando como resultado un valor que nos servirá para entrenar los clasificadores débiles.

A continuación describiremos el algoritmo utilizado para recorrer cada una de las imágenes que conforman nuestro banco de datos, en cada una de ellas se generan subventanas de las cuales se extrajeron las características:

Realizamos una iteración, donde  $m$  es el total de las imágenes,  $A$  es el grupo de configuraciones de subventanas de  $2 \times 2$  y  $B$  son las de  $3 \times 3$ .

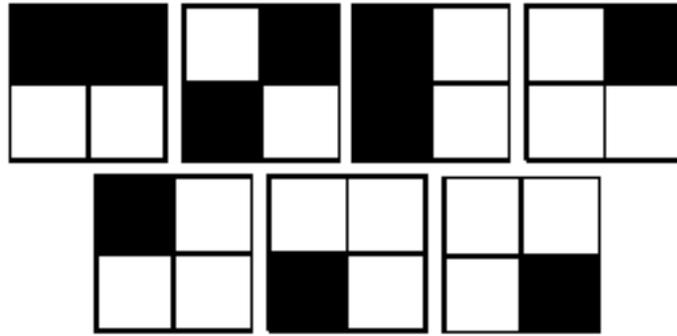


Figura 4.8: Configuraciones para subventanas de  $2 \times 2$ .

1. Para las configuraciones del grupo A, donde:
  - Ancho\_Y= el ancho de la ventana con respecto al eje 'Y'.
  - Ancho\_X= el ancho de la ventana con respecto al eje 'X'.
  - Coordenada\_X= el punto X en la imagen donde se empieza a dibujar el rectángulo.
  - Coordenada\_Y= el punto Y en la imagen donde se empieza a dibujar el rectángulo
2. Se inicializan los valores del Ancho y Coordenada, siendo la ubicación inicial la coordenada (1,1) en los ejes (x,y), (esquina superior izquierda). A partir de ello se construye la subventana empezando por un Ancho\_X de tamaño 2 y un Ancho\_Y de igual manera de tamaño 2, posteriormente se aplica el algoritmo de la sección (seguir sección) para generar las subventanas, con base a ello se prosigue con la extracción de características del grupo A, en donde se extraen las 7 distintas características mostradas de acuerdo a restas y sumas aritméticas de la sumatorias de las regiones representadas por el color blanco o el negro.
3. El procedimiento anterior se repite, pero incrementando primero los puntos de inicio X y Y de 3 en 3 hasta barrer toda la imagen, para

posteriormente incrementar los anchos, primero el X de 4 en 4 y posteriormente el Y, de esa manera se obtiene una amplia cantidad de características con las configuraciones antes mencionadas.

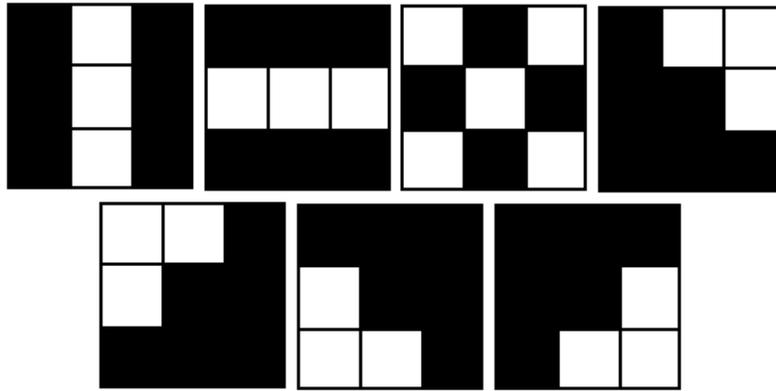


Figura 4.9: Configuraciones para subventanas de 3x3.

1. De igual manera para las configuraciones del grupo B, donde:
  - Ancho\_Y= el ancho de la ventana con respecto al eje 'Y'.
  - Ancho\_X= el ancho de la ventana con respecto al eje 'X'.
  - Coordenada\_X= el punto X en la imagen donde se empieza a dibujar el rectángulo.
  - Coordenada\_Y= el punto Y en la imagen donde se empieza a dibujar el rectángulo
2. Se inicializan los valores del Ancho y Coordenada siendo la ubicación inicial la coordenada (1,1) en los ejes (x,y), (esquina superior izquierda). A partir de ello se construye la subventana empezando por un Ancho\_X de tamaño 2 y un Ancho\_Y de igual manera de tamaño 2, posteriormente se aplica el algoritmo para generar las subventanas de 3x3, con base a ello se prosigue con la extracción de características del

grupo B, en donde se extraen las 6 distintas características mostradas de acuerdo a restas y sumas aritméticas de la sumatorias de las regiones representadas por el color blanco o el negro.

3. El procedimiento anterior se repite, pero incrementando primero los puntos de inicio X y Y de 3 en 3 hasta barrer toda la imagen, para posteriormente incrementar los anchos; primero el X de 3 en 3 y posteriormente el Y, de esa manera se obtiene el resto de las características que conforman nuestros datos.

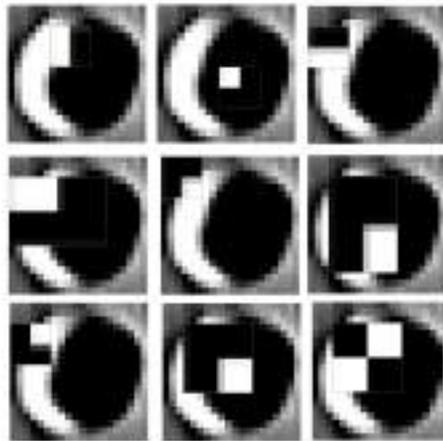


Figura 4.10: Aplicando las distintas plantillas en una muestra de entrenamiento.

Con las características extraídas se generó una matriz bidimensional para los datos sean extraídos de una manera más rápida cuando el algoritmo de “*Adaptive Boosting*” realiza la búsqueda de clasificadores débiles. Cada columna de la matriz representa los distintos componentes del conjunto de entrenamiento, y cada fila representa el resultado de aplicar una plantilla de un tamaño y configuración específico.

## 4.4. Entrenamiento del Modelo

El algoritmo de aprendizaje que utilizamos (“*Adaptive Boosting*”) fue descrito en el capítulo anterior, este entrena el clasificador utilizado para detectar los grupos de píxeles que contienen un cráter. Recordando, sabemos que el algoritmo de “*Adaptive Boosting*” arroja como salida el modelo de un clasificador, el cual se encuentra conformado por múltiples clasificadores débiles:

$$H(x) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x) \right) \quad (4.3)$$

Se estableció que el clasificador final estuviera conformado por 50 clasificadores débiles, a lo cual el algoritmo “*Adaptive Boosting*” construye el clasificador a partir de la suma ponderada de los resultados de los 50 primeros clasificadores débiles seleccionados por este. En este algoritmo ocurren dos procesos, el de selección y el de entrenamiento (cálculo de pesos nuevos en cada iteración del algoritmo y cálculo del peso del clasificador), obteniendo de esa manera el clasificador que utilizamos.

## 4.5. *Support Vector Machines*

De igual manera se utilizó el modelo de entrenamiento de SVM para conseguir un clasificador que podamos utilizar a la par que el entrenado por el “*Adaptive Boosting*”, cabe aclarar que este modelo nos sirvió como punto de comparación para la primera etapa de detección antes del posterior filtrado de las muestras.

Preparar las características que utilizaremos para entrenar el modelo mediante lo SVM, requiere realizar pruebas que se detallan en el capítulo de Resultados, donde los resultados de cada una de las iteraciones son comparados, y con base a eso se eligen los primeros 20 clasificadores que tuvieron mejor rendimiento para la construcción del conjunto de muestras. Los 20 clasificadores nos indicarán cuáles son las características que utilizaremos para

entrenar nuestro modelo principal, ya que cada uno de los clasificadores es concretamente un clasificador por umbral de un mismo grupo de características, y otro clasificador distinto maneja otro conjunto de características. Bajo esa premisa se seleccionan los 20 distintos grupos de características que representan los resultados de aplicar las 20 distintas plantillas Haar a cada una de las imágenes que conforman la base de datos.

Con las 20 distintas características por imagen se forma un vector de características que representan un vector 20-dimensional en donde el algoritmo de SVM buscará un hiperplano que separe de forma óptima los puntos que conforman la clase “Cráter” de la “No cráter”. Aunque ello implicaría proyectar los puntos a un espacio de dimensionalidad superior y de esa manera lograr que el hiperplano tenga una máxima distancia posible de los puntos que se encuentren cerca del mismo.

Algunas clasificaciones binarias no tienen un hiperplano que pueda separar los datos de distintas clases, es por ello que se realiza una variante para poder simplificar la búsqueda de un hiperplano, para ello se realiza la transformación de datos. Tomando en cuenta lo ilustrado en el capítulo anterior, este proceso lo podemos detallar de la siguiente manera:

1. Se selecciona el Kernel a utilizar para mapear el espacio de los datos a un nuevo espacio, donde utilizamos algunos de los Kernels mencionados en el capítulo del marco teórico.

$$F = \{\varphi(x) | (x) \in X\} \quad (4.4)$$

$$x = \{x_1, x_2, x_3, \dots, x_n\} \rightarrow \varphi(x) = \{\varphi(x)_1, \varphi(x)_2, \dots, \varphi(x)_n\}$$

2. Una vez en el espacio de mayor dimensionalidad se busca el hiperplano que separe de manera correcta las muestras.
3. El uso de kernels se respalda por el uso de métodos computacionales de hiperplanos, es por ello que al realizar cálculos que no son más productos puntos. Sin embargo, gracias a ello, finalmente con los datos mapeados a una mayor dimensionalidad, se obtiene un clasificador binario el cual es una hipersuperficie en un espacio  $S$ , el cual determina si un punto nuevo corresponde a cierta clase o no de acuerdo al espacio delimitado que delimita dicha hipersuperficie.

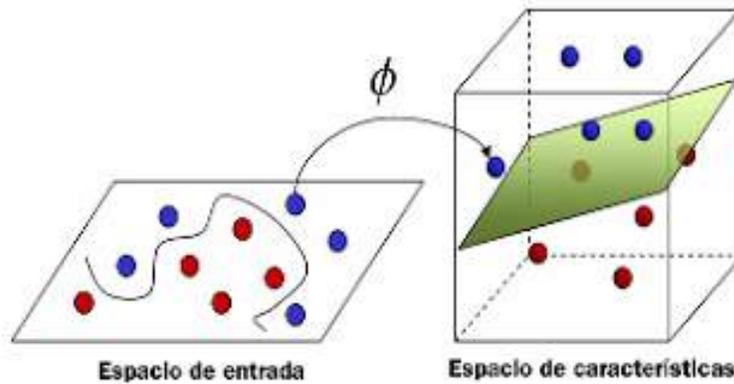


Figura 4.11: Conversión de datos a un espacio de características.

## 4.6. Filtro de Dirección de la Luz

Posteriormente, luego de la clasificar las muestras con el clasificador entrenado ya sea por el método de “*Adaptive Boosting*” o de SVM, se prosiguió con un sistema de filtros para eliminar de esa manera las muestras detectadas erróneamente y aumentar la eficacia del sistema. Es por ello que a cada muestra identificada por el “*Adaptive Boosting*” o el SVM se le aplicó un sistema de filtros empezando por el filtro de la dirección de la luz.

Como primer paso se obtuvo la dirección de la luz de la muestra, para que de esa manera de forma estadística supiéramos si correspondía a un cráter o a una formación topográfica distinta. Este punto se propone debido a que en una imagen la dirección de la luz es una misma y por ello las proyecciones que tienen los cráteres siguen un mismo patrón que varía levemente de acuerdo a la topografía y trazado de su borde, en cambio en formaciones montañosas o en bordes la dirección de la proyección difiere de la estadística mostrada por la dirección que tienen los cráteres. Con base a esa observación se calculó la dirección de la muestra detectada para etiquetar todas y posteriormente eliminar aquellas que no cumplan con los parámetros establecidos.

### 4.6.1. Calculando la Dirección de la Luz

Para el cálculo de la luz se estableció una estadística para agrupar los píxeles claros en una región, así como los oscuros en otra región, cabe aclarar que los parámetros utilizados fueron los siguientes:

- Para una imagen  $K$ , los píxeles se etiquetan como “oscuros” y se agrupan si son aquellos que conforman el intervalo  $A \geq x \geq B$ .  
Donde  $A$  es el valor del píxel con menor intensidad encontrado en la imagen  $K$  y  $B$  es el umbral establecido de tal manera que la distancia entre  $A$  y  $B$  sea la misma que  $B$ , y el valor promedio de los píxeles que conforman la imagen  $K$ .
- Para la misma imagen  $K$ , los píxeles cuyo valor se encuentra en el intervalo  $J \geq x \geq Y$  son etiquetados como “claros” y se agrupan para el posterior análisis.  
Donde  $Y$  es valor del píxel con mayor intensidad presente en la imagen  $K$  y  $J$  es el umbral establecido para que la distancia entre el promedio de los valores a ese umbral sea igual al de dicho umbral al valor  $Y$ .

Cabe aclarar que los valores  $A$ ,  $B$ ,  $K$ ,  $J$  son distintos para cada muestra, de esa manera se garantiza que para toda muestra que entra en este filtrado se haga un análisis sin establecer valores predeterminados y basándose en los valores mínimos, máximos y el promedio de los valores que presentan los píxeles que conforman dicha muestra.

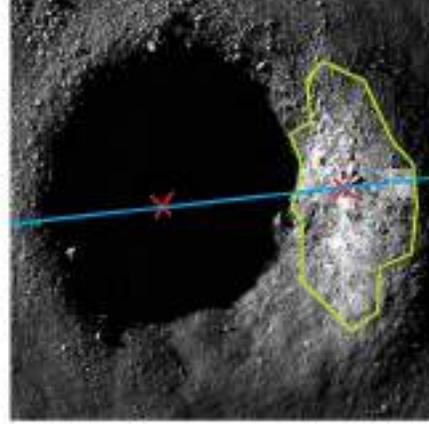


Figura 4.12: Calculo de la dirección de la luz de un cráter.

Posteriormente, ya con los grupos conformados se calcularon los centroides de dichas regiones para de esa manera trazar un vector, para ello se siguió el siguiente algoritmo propuesto:

Para cada una de las muestras se hizo lo siguiente:

1. Recorrimos cada uno de los píxeles para almacenar la ubicación de los píxeles que pertenecen a la región clara o la oscura con base a los parámetros obtenidos para esa muestra.
2. Al ser un problema discreto debido a la división de la imagen en píxeles, obtuvimos los centroides de las 2 regiones recorriendo ambos conjuntos de píxeles de acuerdo a las siguientes fórmulas:

$$c_{x1} = \frac{1}{N1} \sum_{i=1}^N 1x_i \quad c_{y1} = \frac{1}{N1} \sum_{i=1}^N 1y_i \quad (4.5)$$

$$c_{x2} = \frac{1}{N2} \sum_{i=1}^N 2x_i \quad c_{y2} = \frac{1}{N2} \sum_{i=1}^N 2y_i \quad (4.6)$$

Donde  $c_{x1}$  y  $c_{y1}$  son las coordenadas  $(c_{x1}, c_{y1})$  del centroide de la región clara que obtenemos al recorrer todos los  $N1$  píxeles de dicha región, y

$c_{x2}$  y  $c_{y2}$  son las coordenadas  $(c_{x2}, c_{y2})$  del centroide de la región oscura que se encuentra conformada por los  $N2$  píxeles.

3. Posteriormente se calcula el ángulo del vector-línea que une estos dos centroides el cual nos indica la proyección que tienen los colores oscuros sobre la imagen, el valor de los grados es almacenado para el posterior análisis estadístico de todas las muestras. El cálculo de los grados se hace de la siguiente manera:

$$\theta = \tan^{-1} \left( \frac{c_{y2} - c_{y1}}{c_{x2} - c_{x1}} \right) \quad (4.7)$$

## 4.7. Filtro de Uniformidad de la Subventana

El segundo filtro consistió en realizar una inspección en la imagen para descartar aquellas que no tengan presencia de regiones oscuras y claras, ya que corresponden a zonas que a pesar de no contar con cráteres lunares y ser demasiado oscuras o luminosas fueron identificadas por el clasificador como posibles cráteres.

Este filtrado lo que hace básicamente es un conteo de los píxeles que conforman las regiones claras y las regiones muy oscuras partiendo de las definiciones expresadas en la sección de filtro de dirección. Como primer paso se recorrió la imagen  $K$  y se realizó el conteo de los píxeles de las regiones mencionadas.

Siendo  $rc$  el número de píxeles que conforman la región clara y  $rh$  la región oscura,  $i$  las coordenadas en 'x' dentro la imagen  $K$  y  $j$  las coordenadas en 'y'.

```

rc=0;
rh=0;
TamanoX=24;
TamanonoY=24;
i=0;
j=0;
while  $i \leq TamanoX$  do
  while  $j \leq TamanonoY$  do
    if  $K(i, j) == claro$  then
      | rc=rc+1;
    end
    if  $K(i, j) == oscuro$  then
      | rh=rh+1;
    end
  end
end

```

Obteniendo de esa manera un número total de píxeles que conforman una u otra zona. Recordando, establecimos como claros los píxeles cuyo valor se encuentran en el rango  $J \geq x \geq Y$  y oscuros en el rango  $A \geq x \geq B$ . Si el número total de píxeles  $rc$  no cumple la siguiente condición descartamos la muestra:

$$rc \geq \theta (tamanoY + tamanoX) \quad (4.8)$$

Donde  $\theta$  es un parámetro establecido en este caso establecimos que las regiones sean al menos el 15 % de la imagen, por ende  $\theta = \frac{1}{10}$ .

De igual manera si el número total de píxeles  $rh$  no cumple la siguiente condición descartamos la muestra:

$$hc \geq \theta_2 (tamanoY + tamanoX) \quad (4.9)$$

Donde  $\theta_2$  es otro parámetro establecido en este caso establecimos que las regiones sean al menos el 15 % de la imagen, por ende  $\theta = \frac{1}{5}$ .



Figura 4.13: El cálculo de uniformidad permite descartar muestras.

## 4.8. Filtrando las Muestras por Grados

Aparentemente solo realizamos un filtrado de la muestra, al rechazar aquellas que no pasen el filtrado de uniformidad, ya que al realizar el cálculo de los grados no se rechazaron muestras, sin embargo al conocer la proyección que tienen las sombras en una imagen se puede establecer un umbral  $\theta = 24$  para filtrar las muestras.

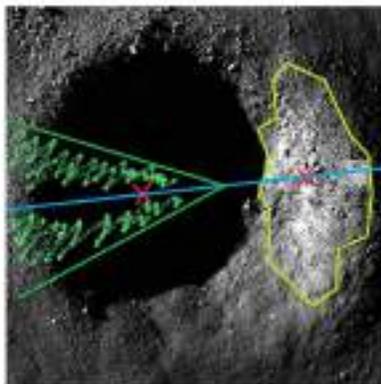


Figura 4.14: Umbral de 24 grados establecido para filtrar el resto de las muestras.

Para establecer el umbral al aplicar el detector de cráteres como observaremos a continuación, se recurrió a la estadística para de esa manera obtener ese umbral  $\theta$  que englobe a la mayoría de las muestras y deseche a las que corresponden a formaciones topográficas o bordes de cráteres que fueron detectados como uno solo cráter.

1. Como primer paseo se utilizó el algoritmo para obtener histogramas y de esa manera sacar la distribución de los cráteres presentes, discretizando los anteriores para de esa manera tenerlos repartidos en 360 distintos grados, obteniendo de esa manera una gráfica de histogramas.

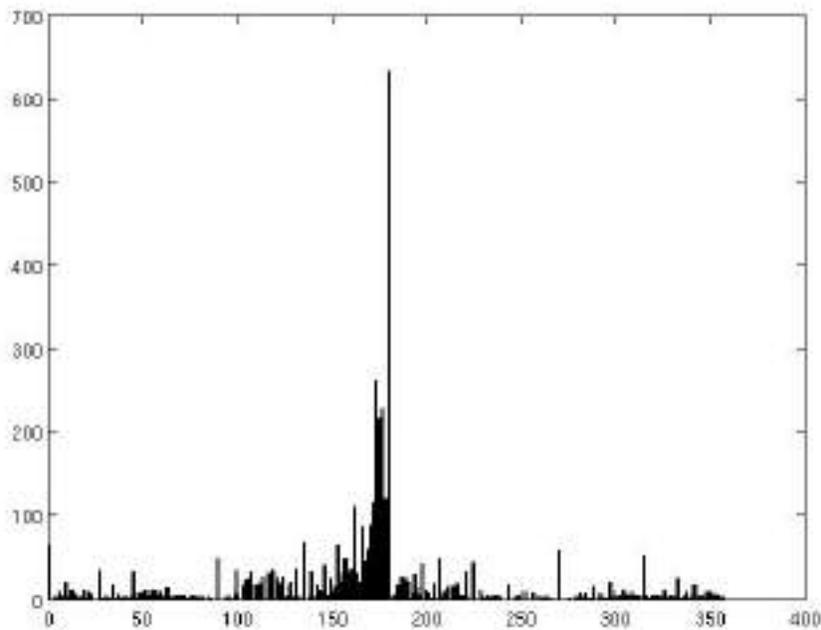


Figura 4.15: Histograma de dirección de la luz de un grupo de muestras detectadas por el clasificador en cuestión.

2. Posteriormente, se seleccionó aquel rango de 24 grados de ancho en donde se agrupaban la mayor cantidad de muestras.

3. Con base al rango seleccionado, se selecciona el umbral  $\theta$  y se establecen los rangos que se utilizarán para el filtrado.

# Capítulo 5

## Experimentos y Resultados

En este capítulo, se describirán las pruebas que se realizaron para la evaluación del sistema propuesto, para la evaluación se utilizó la validación cruzada de  $k$  iteraciones siendo  $k = 10$  con distintos clasificadores entrenados poniendo a prueba estos y comparando los resultados entre sí, cabe señalar que los clasificadores a probar son 4: siendo uno entrenado por “*Adaptive Boosting*” utilizando las plantillas originales, el otro con las plantillas propuestas. El tercer clasificador es entrenado por “*SVM*” con un kernel polinomial-homogéneo y por último, el cuarto se utiliza igualmente “*SVM*” con un kernel gaussiano radial, ambos utilizando las plantillas propuestas en este trabajo.

Posteriormente se mostrarán los resultados obtenidos al aplicar el sistema de detección propuesto en varias imágenes de superficie lunares para evaluar los métodos de detección en distintas áreas de la superficie lunar. Para la evaluación se compararán los cráteres detectados automáticamente a una imagen de referencia con cráteres detectados manualmente.

Para las pruebas se utilizó el banco de imágenes que fue extraído por el método descrito en el capítulo de la metodología, teniendo un total de 960 muestras negativas y 960 muestras positivas de las cuales se obtendrán los subconjuntos para las pruebas que se realizarán.

## 5.1. Validación Cruzada de $k$ Iteraciones

La validación cruzada es un método estadístico empleado para la evaluación y comparación de distintos clasificadores. Este método consiste en la división de un conjunto de muestras en varios subconjuntos de entrenamiento y de prueba, posteriormente cada uno de los subconjuntos obtenidos es evaluado de manera individual, y el promedio de dichas evaluaciones nos dan una referencia del desempeño de nuestro algoritmo.

Una de las formas básicas de validación cruzada es por  $k$  iteraciones, en donde el conjunto se divide en  $k$  subconjuntos cada una con el mismo número de elementos de las clases correspondientes, para a partir de ellos formar nuevos conjuntos de entrenamiento y prueba. A continuación detallaremos el procedimiento a seguir:

1. Al azar elegimos sin repetir los elementos que conformarán los 10 subconjuntos de datos, estos deberán estar conformados por partes iguales de tanto muestras negativas, como muestras positivas. Al ser 960 muestras de cada clase se seleccionan 96 muestras positivas y 96 muestras negativas para cada subconjunto de muestras.
2. El siguiente paso consiste en la creación de los conjuntos de prueba y entrenamiento para la validación, para ello se selecciona 1 de los subconjuntos creados en el paso anterior y ese se establece como conjunto de prueba para la iteración  $k$  y los otros conjuntos en este caso 9; debido a que  $k = 10$ , se unen formando el conjunto de entrenamiento. Ese procedimiento se repite  $k - 1$  veces formando los 10 grupos distintos de prueba y entrenamiento de tal manera que cada uno de los 10 subconjuntos creados anteriormente son utilizados para las pruebas.
3. El siguiente paso en la validación cruzada de  $k$  iteraciones consiste en la evaluación de cada una de las muestras preparadas en donde se entrena con el conjunto 1 y se prueba con su equivalente y así sucesivamente con los 10 distintos grupos.
4. Al final obtenemos un resultado que mide efectividad para cada uno de los grupos. Este resultado se promedia para obtener la evaluación del clasificador que estamos utilizando.

El procedimiento anterior se aplica para los 4 distintos clasificadores probados para su comparación entre ellos.

## 5.2. Sensibilidad y Especificidad

Para los comparativos entre las distintas iteraciones se utilizaron los indicadores de sensibilidad y especificidad [38] para de esa manera evaluar los distintos clasificadores entrenados ya que cada clasificador varía sus parámetros con respecto a los demás, en función de las muestras utilizadas como conjunto de entrenamiento. Definimos como sensibilidad a la probabilidad que tiene el clasificador de considerar una muestra como positiva cuando realmente es positivo.

$$s = \frac{VP}{VP + FN} \quad (5.1)$$

Donde  $s$  es la sensibilidad,  $VP$  representa el número de muestras detectadas como positivos de manera correcta y  $FN$  el número de ejemplos que fueron detectados como negativos a pesar de tener una etiqueta de comparación de ser positivos (etiquetados erróneamente negativos), lo cual es denominado como falso negativo.

Otros de los indicadores principales para la evaluación de los clasificadores es la especificidad. Nos referimos como especificidad a la probabilidad de que un ejemplo sea considerado como negativo dado que realmente es negativo según las etiquetas otorgadas previamente.

$$e = \frac{VN}{VN + FP} \quad (5.2)$$

Donde  $e$  es la especificidad, y de igual manera observamos que  $VN$  son las muestras negativas detectadas correctamente como negativas y  $FP$  son aquellas muestras etiquetadas como positivas erróneamente (falsos positivos).

### 5.3. Evaluación de los algoritmos

Para entrenar y probar los algoritmos se utilizaron las imágenes previamente mencionadas de la misión Apollo las cuales fueron extraídas de unas pocas imágenes las cuales presentan la misma dirección de luz y posteriormente rotadas hasta obtener un total de 960 muestras positivas y 960 muestras negativas.

#### 5.3.1. Resultados de la Validación Cruzada

A continuación mostraremos los resultados obtenidos por las  $k$ -iteraciones con los 4 distintos clasificadores a aplicarse. Para cada uno de ellos se realizó una validación cruzada y el resultado obtenido nos ayudó a comparar las diferencias entre ellos.

##### **Validación Cruzada con Clasificadores Entrenados con “*Adaptive Boosting*”**

Como primer prueba se realizó el entrenamiento de las muestras con el conjunto de características obtenidas al aplicar las plantillas Haar propuestas en [23] para de esa manera probar el desempeño utilizando las plantillas originales. Para ello se aplicó la validación cruzada de 10 elementos como mencionamos previamente, dividiendo los grupos en 10 particiones iguales y formando 10 subconjuntos. En esta prueba podemos observar que se obtiene un buen porcentaje de detección con una sensibilidad de 0.9250 y una especificidad de 0.9400, cuando el modelo llega a los 30 clasificadores. Estos valores ya no mejoran de manera significativa al aumentar el número de clasificadores débiles ya que la curva del error se estabiliza como podemos observar en la figura 5.1.

En las siguientes tablas 5.1 y 5.2 observamos la detección de los negativos falsos para 6, 15 y 30 elementos y de igual manera los positivos falsos para igual número de muestras. Cabe aclarar que un “falso negativo”, es aquella muestra que fue etiquetada como negativa durante el proceso pero en realidad

es una muestra positiva. De igual manera llamamos como “falso positivo” a aquella muestras etiquetadas como positivas, pero que son negativas. Tener un indicador de ello permite conocer el rendimiento del algoritmo.

Partición	Falsos Negativos		
	6	15	30
P1	0.35	0.225	0.05
P2	0.375	0.2	0.05
P3	0.4	0.175	0.15
P4	0.3	0.175	0.075
P5	0.025	0.075	0.075
P6	0.2	0	0
P7	0.25	0.075	0.025
P8	0.3	0.075	0.175
P9	0.3	0.15	0.1
P10	0.275	0.04	0.05
Media	0.277	0.119	0.075
$\sigma$	0.106	0.075	0.054

Tabla 5.1: Desempeño de los vectores de características generados al aplicar plantillas de [23]. Evaluando detecciones Negativas erróneas.

Partición	Falsos Positivos		
	6	15	30
P1	0.025	0.05	0.025
P2	0.025	0	0.025
P3	0	0	0
P4	0.025	0.025	0.05
P5	0.125	0.05	0.075
P6	0.275	0.175	0.25
P7	0.125	0.1	0
P8	0.175	0.225	0.025
P9	0.175	0.025	0
P10	0.15	0.2	0.15
Media	0.11	0.085	0.06
$\sigma$	0.089	0.085	0.08

Tabla 5.2: Desempeño de los vectores de características generados al aplicar plantillas de [23]. Evaluando detecciones Positivas erróneas.

Sensibilidad	0.9250
Especificidad	0.9400

Tabla 5.3: Sensibilidad y Especificidad para Clasificador entrenado con características extraídas de plantillas genéricas.

Como observamos, para el clasificador utilizando las plantillas tradicionales, se obtuvo una media para negativos falsos de 0.075 y una para los positivos falsos de 0.06 con desviaciones estándar de 0.054 y 0.08 respectivamente, en la siguiente figura observaremos como el error fue disminuyendo en el clasificador fuerte creado hasta que se estabiliza y no hay cambios significativos.

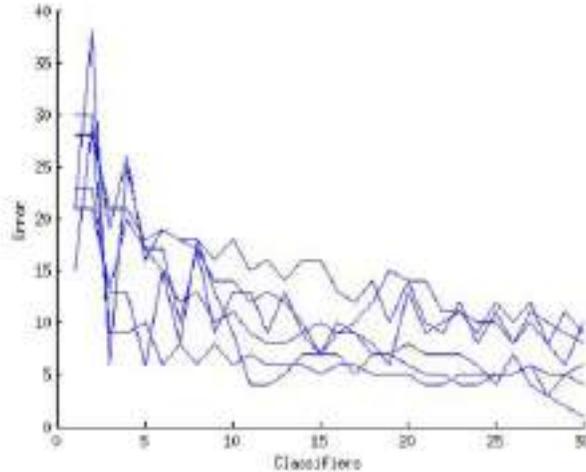


Figura 5.1: Gráfica del decremento del error al ir aumentando el número de clasificadores.

Posteriormente se realizó el mismo experimento pero con las plantillas propuestas, con la diferencia que se incrementó el alcance de las características como podemos observar ya que se aumentó el número de clasificadores hasta 50. Esa determinación se dio ya que en la mayoría de las muestras se tuvo un mejor rendimiento con estas plantillas y se determinó ampliar el número de características en esta prueba, para de esa manera observar el decaimiento de la gráfica de entrenamiento vs error que observamos en la gráfica 5.2.

Las pruebas realizadas mostraron una media para “Falsos Positivos” de 0.0893, 0.0891 y 0.0583 y de “Falsos Negativos” de 0.0786, 0.0581 y 0.0426 para 15, 25 y 50 clasificadores respectivamente. Teniendo unas desviaciones estándares de 0.0862, 0.1135, 0.1093 y 0.0812, 0.0724, 0.0704 respectivamente. Pudimos observar un mejor rendimiento y la gráfica muestra una cercanía mayor a los cero errores al llegar al rango de los 50 clasificadores mostrando un funcionamiento similar al anterior clasificador pero con la ventaja de que al tener mayor información, el algoritmo se “especializa” más en esos cráteres.

Partición	Falsos Positivos		
	15	25	50
P1	0.05	0.031	0.031
P2	0.073	0.021	0.0104
P3	0.1145	0.073	0.0521
P4	0.031	0.0104	0
P5	0.083	0.083	0.0521
P6	0.0625	0.0312	0.0104
P7	0.0416	0.0312	0.0104
P8	0.031	0.031	0.0104
P9	0.3229	0.4375	0.3646
P10	0.0833	0.0833	0.0416
Media	0.0893	0.0891	0.0583
$\sigma$	0.0862	0.1135	0.1093

Tabla 5.4: Desempeño de los vectores de características generados al aplicar plantillas propuestas. Evaluando detecciones Positivas erróneas.

Partición	Falsos Negativos		
	15	25	50
P1	0.06	0.031	0.0104
P2	0.083	0.0521	0.031
P3	0	0	0
P4	0.0104	0.0104	0.031
P5	0	0	0
P6	0.166	0.0625	0.0208
P7	0.2604	0.25	0.239
P8	0.06	0.06	0.031
P9	0.0937	0.0729	0.0416
P10	0.0521	0.0416	0.021
Media	0.0786	0.0581	0.0426
$\sigma$	0.0812	0.0724	0.0704

Tabla 5.5: Desempeño de los vectores de características generados al aplicar plantillas propuestas. Evaluando detecciones Negativas erróneas.

Sensibilidad	0.9574
Especificidad	0.9417

Tabla 5.6: Sensibilidad y Especificidad para Clasificador entrenado con características propuestas.

Podemos observar que se los valores de sensibilidad y especificidad incrementan comparados con el anterior clasificador, aumentando a 0.9574 y 0.9417 respectivamente. A continuación se mostrarán los resultados obtenidos de este clasificador entrenado con las plantillas propuestas:

Resultados Adaptive Boosting			
Partición	15	25	50
P1	94.5	96.9	97.93
P2	92.22	96.345	97.93
P3	94.275	96.35	97.395
P4	97.93	98.96	98.45
P5	95.85	95.85	97.39
P6	88.575	95.32	98.44
P7	84.9	85.94	87.53
P8	95.45	95.45	97.93
P9	79.17	74.48	79.7
P10	93.23	93.76	96.87
Media	91.6	92.64	94.96
$\sigma$	.0423	.0972	.0861

Tabla 5.7: Tasa de detección alcanzada por el clasificador entrenado con Adaptive Boosting para cada una de las particiones.

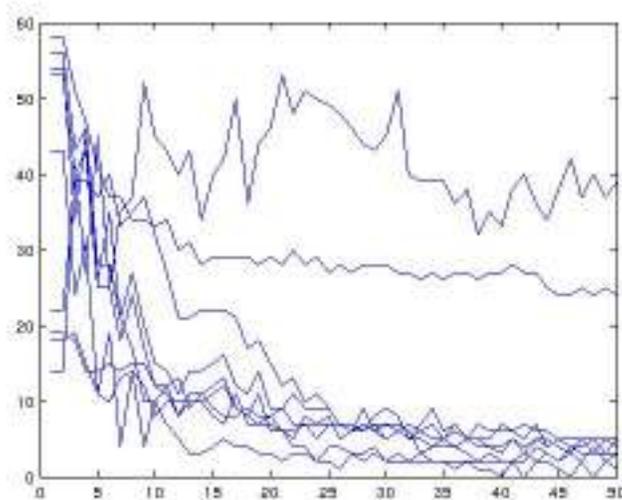


Figura 5.2: Gráfica del decrecimiento del error al ir aumentando el número de clasificadores. Observamos que aun teniendo un mejor desempeño, dos de los grupos tuvieron más errores.

Este clasificador mostró varias muestras que, al ir analizando su origen antes de ser rotadas correspondieron a distintas muestras de cráteres que eran muy tenues comparados con los demás o en su defecto representaban a muestras de bordes de cráteres. En la siguiente imagen observaremos 5 ejemplos de cada tipo.

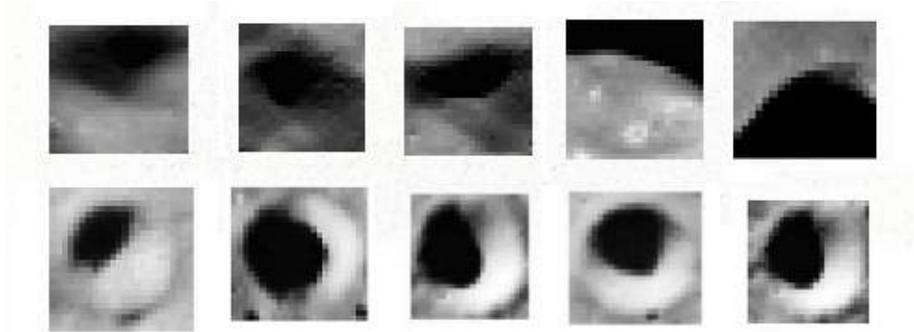


Figura 5.3: Ejemplos de muestras catalogadas erróneamente. En la fila superior observamos “Falsos Positivos” y en fila inferior “Falsos Negativos”.

### Validación Cruzada con Clasificadores Entrenados con “SVM”

Para comparar los resultados obtenidos de igual manera se entrenaron dos clasificadores utilizando el algoritmo de *Support Vector Machines*. Los dos clasificadores tuvieron distinto Kernel, siendo el primero de tipo gaussiano radial y el segundo de tipo polinomial de orden 3. Para ello se eligieron las características que conformaron la matriz de muestras, se seleccionaron las 20 distintas plantillas que corresponden a las 20 primeros clasificadores débiles del clasificador que perteneció a una de las particiones que tuvieron mejor rendimiento, siendo la número 5. Con esas plantillas se determinó que características se utilizaron para conformar los vectores de características de cada una de las muestras. Al haber sido calculadas previamente simplemente se seleccionaron del conjunto de muestras global. En el primero de tipo gaussiano radial obtuvo una media de 0.2010 para “Falsos Negativos” y de 0.0062 para “Falsos Positivos”, con desviaciones estándares de 0.1128 y 0.0112 respectivamente. Podemos observar que tiene un alto desempeño en cuanto a especificidad (0.9938) y un aceptable 0.7900 en cuanto a sensibilidad sin ser mayor que el clasificador donde se utilizó el modelo de *Adaptive Boosting*. Esto nos muestra que si bien este algoritmo es bueno para etiquetar las muestras negativas, falla en cuanto a rendimiento descartando demasiadas muestras positivas para lo que buscamos.

Partición	FN	FP
P1	0.2708	0
P2	0.48	0
P3	0.1246	0.0104
P4	0.0833	0.0312
P5	0.1344	0
P6	0.2396	0
P7	0.1875	0
P8	0.1458	0
P9	0.1979	0.0208
P10	0.1458	0
Media	0.2010	0.0062
$\sigma$	0.1128	0.0112

Tabla 5.8: Desempeño del clasificador *SVM* con un kernel gaussiano radial.

Sensibilidad	0.7900
Especificidad	0.9938

Tabla 5.9: Sensibilidad y Especificidad para Clasificador entrenado con características propuestas.

En el segundo de tipo polinomial de orden 3 obtuvo una media de 0.1354 para “Falsos Negativos” y de 0.0417 para “Falsos Positivo ’ ’”, con desviaciones estándares de 0.2026 y 0.0250 respectivamente. Podemos observar que tiene un incremento en cuanto a su desempeño con los valores de especificidad (0.9583) y de sensibilidad (0.8646), siendo este de mejor rendimiento que el anterior. Este último fue utilizado para las pruebas posteriores en imágenes de superficies planetarias.

Partición	FN	FP
P1	0.0521	0.0312
P2	0.1667	0.0625
P3	0.0104	0.0312
P4	0.0312	0.0417
P5	0	0.0312
P6	0.6879	0
P7	0.1354	0.0833
P8	0.0833	0.0417
P9	0.0417	0.0729
P10	0.1458	0.0208
Media	0.1354	0.0417
$\sigma$	0.2026	0.0250

Tabla 5.10: Desempeño del clasificador *SVM* con un kernel polinomial de orden 3.

Sensibilidad	0.8646
Especificidad	0.9583

Tabla 5.11: Sensibilidad y Especificidad para Clasificador entrenado con características propuestas.

De igual manera podemos concluir que si bien los clasificadores utilizados son suficientes para tener un mejor rendimiento, posiblemente no son los adecuados para este tipo de algoritmo. En la siguiente tabla, observaremos el desempeño de ambos modelos para las particiones formadas:

Partición	SVM 1	SVM 2
P1	86.46	95.84
P2	76	88.54
P3	93.25	97.92
P4	94.27	96.35
P5	93.28	98.44
P6	88.02	65.60
P7	90.63	89.06
P8	92.71	93.75
P9	89.06	94.27
P10	92.71	91.67
Media	89.64	91.15
$\sigma$	0.112	0.20235

Tabla 5.12: Desempeño del clasificador entrenado por SVM en cada una de las particiones.

### 5.3.2. Método para Detección de Cráteres Lunares

En esta sección se detallará el proceso utilizado para la detección de cráteres lunares con los 2 clasificadores finales utilizados, en primera instancia se eligió para cada uno aquel que detector de mejor manera su subconjunto en el proceso de validación cruzada. Este clasificador es utilizado para detectar si el grupo de píxeles que el algoritmo selecciona automáticamente es un cráter o forma parte del grupo de las demás formaciones topográficas.

El método consiste en recorrer la imagen y cada una de ellas introducirlas como muestra al clasificador, y si es detectada, pasa por una serie de filtros para finalmente ser etiquetada como zona de posible cráter, posteriormente se realiza un filtrado para evitar que un mismo cráter sea etiquetado dos veces. A continuación se detallará el algoritmo utilizado para la primera etapa de filtrado, así como el proceso para filtrar las múltiples detecciones de un mismo cráter.

Como primer paso, previo al barrido en la imagen donde se realizó la

detección de los cráteres, se calculó la imagen integral de toda la imagen para de esa manera poder acceder a la información requerida cuando el clasificador de “*Adaptive Boosting*” lo requiera, logrando que solo se calcule una sola vez la sumatoria de los píxeles y que se pueda acceder posteriormente a los valores.

Posteriormente se creó una subventana la cual recorre la imagen extrayendo un grupo de píxeles. Dicho grupo de píxeles será analizado por el clasificador elegido. Se estableció un tamaño de ventana mínimo  $C$ , en este caso de  $6 \times 6$  píxeles, el cual se fue incrementando al acabar cada barrida.

1. Se establecen como coordenadas iniciales  $x = 1$  y  $y = 1$ .
2. Se traza la subventana de  $C \times C$  empezando por las coordenadas establecidas  $x$  y  $y$  hacia la derecha y abajo.
3. Se extraen los píxeles contenidos por la subventana, y se guardan como una imagen independiente.
4. Este paso es realizado en función del clasificador elegido. Se introduce el grupo de píxeles al clasificador elegido, este arroja una etiqueta dependiendo de lo que reconoció.
  - Si se eligió un clasificador basado en “*Adaptive Boosting*”, se seleccionan las  $n$  características necesarias para poder introducirlas al clasificador. Y a partir de ellas se evaluaron las características para asignar la etiqueta correspondiente
  - De lo contrario se extraen las características establecidas previamente en el entrenamiento del clasificador “*SVM*”. Y posteriormente se asigna una clase en base a su ubicación con respecto al hiperplano calculado.
5. Si el grupo de píxeles es catalogado como posible cráter, se realizó la siguiente subrutina:
  - Se realizó el cálculo de la dirección de la luz de ese grupo de píxeles y se guardó en un vector de valores.
  - Se guardaron las coordenadas de la ubicación del grupo de píxeles candidatos.

- Se calculó el grado de uniformidad del grupo de píxeles. En este paso, si el grupo de píxeles no presenta la proporción de valores esperada, la etiqueta de “candidato” es descartada, eliminando el “cráter” de la lista de los cráteres detectados.
6. Posteriormente se incrementan las coordenadas de inicio  $x$  y  $y$  hasta recorrer toda la imagen. Una vez finalizado, se aumenta el tamaño establecido  $C$ , multiplicando el tamaño anterior por 1.8, y se reinician las coordenadas de inicio de la subventana  $(x, y)$ . Se estableció un tamaño máximo de 80 píxeles en las pruebas realizadas.

Una vez recorrida toda la imagen, se prosiguió con la eliminación estadística de los “cráteres” que no cumplieron con el filtrado de la dirección de la luz, y de igual manera se realizó un ajuste de los “cráteres” detectados para mostrar solo una vez aquellos que fueron detectados con distintas subventanas, para ello se siguió el siguiente proceso:

1. Se realizó un recorrido de todos los cráteres identificados por el clasificador creando una estructura de datos, donde se guardan los vecinos de cada uno de los cráteres detectados, para ello se establece una condición, en donde un cráter vecino debe estar alejado al menos la mitad del tamaño de la subventana que lo detectó, otra de las condiciones establecidas es que los vecinos solo puedan ser los cráteres que cumplan con la siguiente condición:

$$Tamano_1 * 1,8 \Rightarrow abs(Tamano_1 - Tamano_2) \quad (5.3)$$

Haciendo posible que solo se puedan etiquetar como vecinos aquellos cráteres con un tamaño cercano.

2. El siguiente paso consistió en encontrar los valores de coordenadas mínimos y máximos, recorriendo todos los nodos para cada uno de los cráteres. De esa manera se obtienen las coordenadas  $(x_1, y_1)$ ,  $(x_2, y_1)$ ,  $(x_1, y_2)$  y  $(x_2, y_2)$  las cuales delimitan el área del cráter detectado. Este último paso fue comparado de igual manera con otro método donde se selecciona el primer cráter etiquetado y no se selecciona ningún otro a  $Tamano/2$  distancia de sus bordes. En donde eliminamos aquellos

cráteres identificados a menos de una distancia establecida; ya que presumiblemente son el mismo cráter, de igual manera se limpia la detección presentando una selección de mayor área para cada uno de los cráteres.

En las figuras a continuación se mostrarán los resultados obtenidos para 2 distintas imágenes utilizando un clasificador entrenado conformado por 50 clasificadores débiles utilizando el modelo de entrenamiento propuesto y el otro clasificador utilizado es el “SVM” polinomial de orden 3.

### Resultados del Método de Detección de Cráteres

Para probar los algoritmos se eligieron 2 imágenes de la superficie lunar, una conteniendo una zona llena de cráteres lunares y otra que representa el mar de la humedad, cabe aclarar que para cada una de ellas se utilizaron tanto el clasificador entrenado por el algoritmo de “*Adaptive Boosting*”, cómo el que es entrenado mediante SVM. De igual manera para cada una de ellas se probaron los dos tipos de filtrado espacial propuestos.

A continuación se muestra la imagen de prueba 1, la cual fue etiquetada manualmente para mostrar la salida esperada, cabe aclarar que no se etiquetaron los cráteres que tienen un tamaño menor a 5x5 píxeles. Esta medida se tomó debido a la dificultad de detectar esos cráteres influyó en el entrenamiento de los clasificadores, por ello se tomó la determinación de establecer un tamaño mínimo.

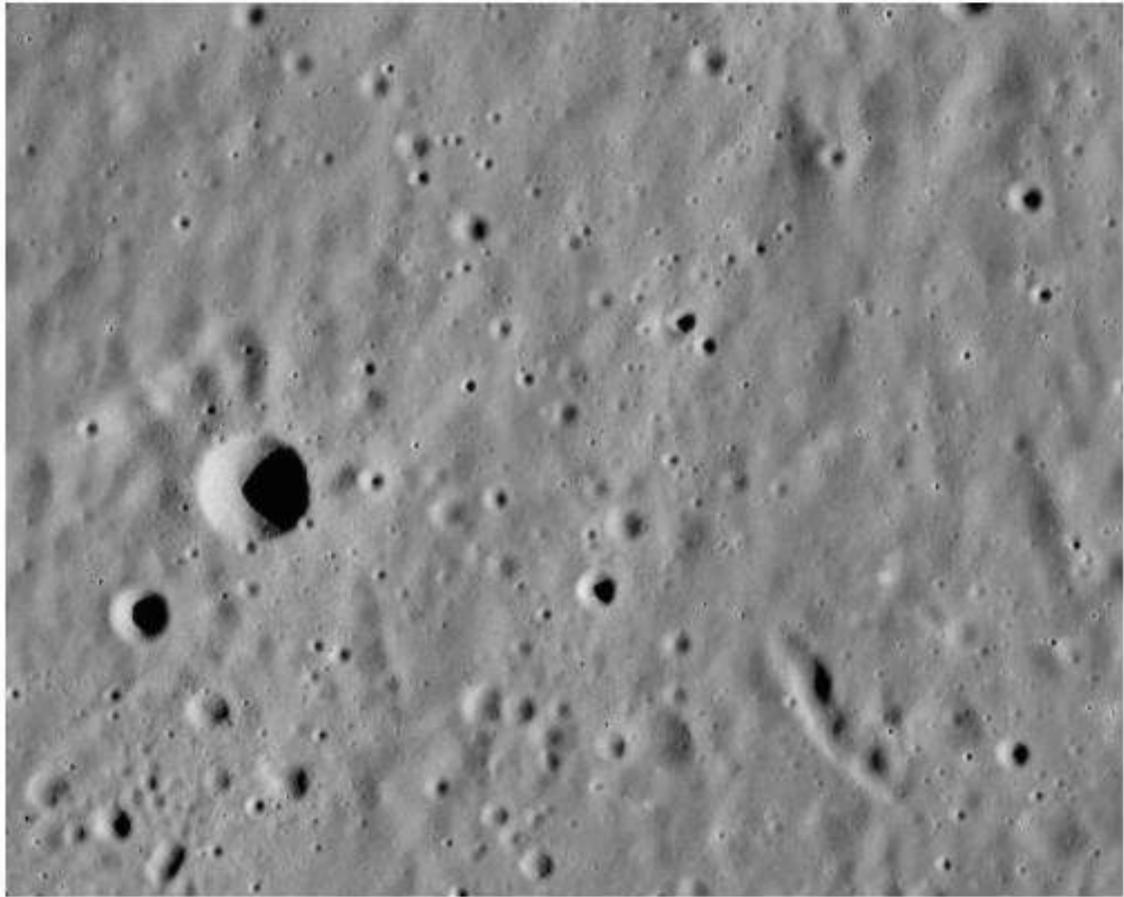


Figura 5.4: Imagen de la superficie lunar, extraída del banco de imágenes de la misión Apollo.

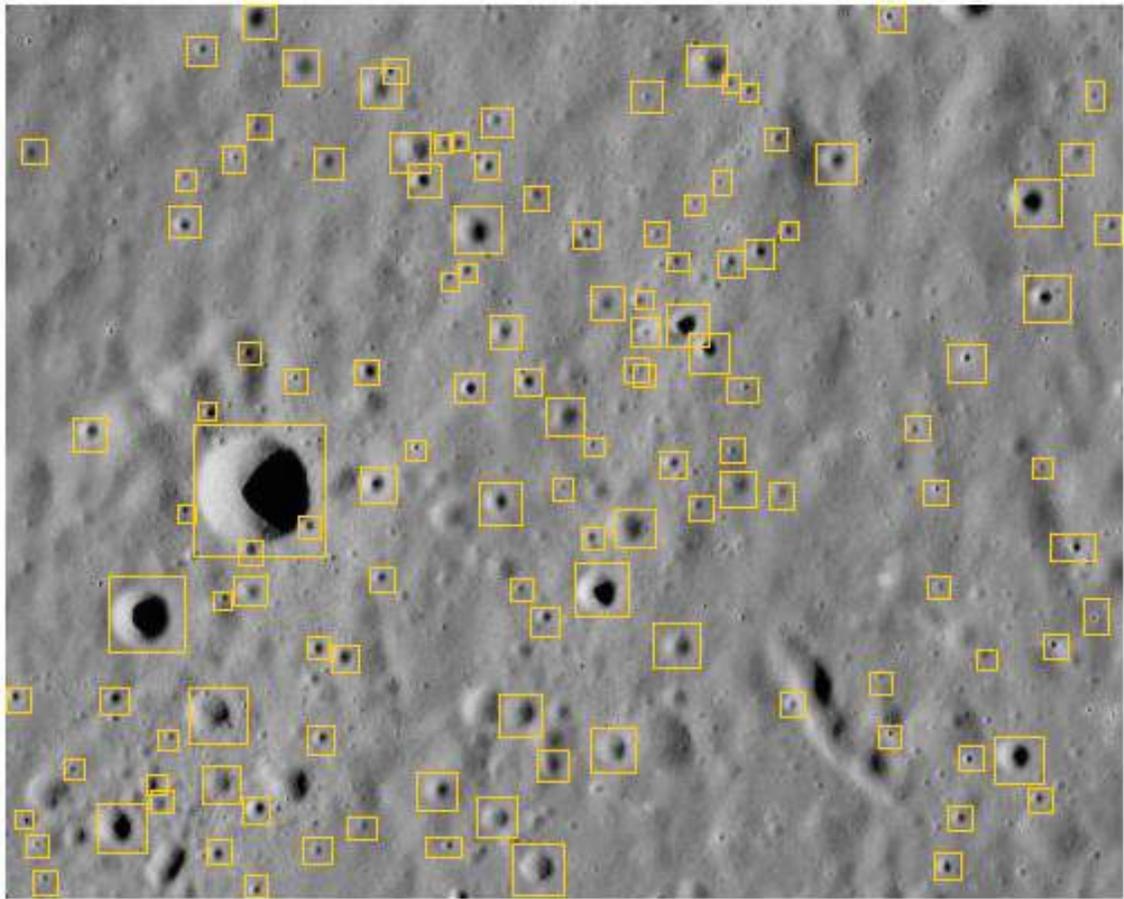


Figura 5.5: Etiquetado manual de cráteres presentes en la imagen de prueba 1.

En las siguientes figuras observamos como el clasificador de “*Adaptive Boosting*” engloba mejor los cráteres teniendo un mejor desempeño en general que el que utilizan las SVM en este caso.

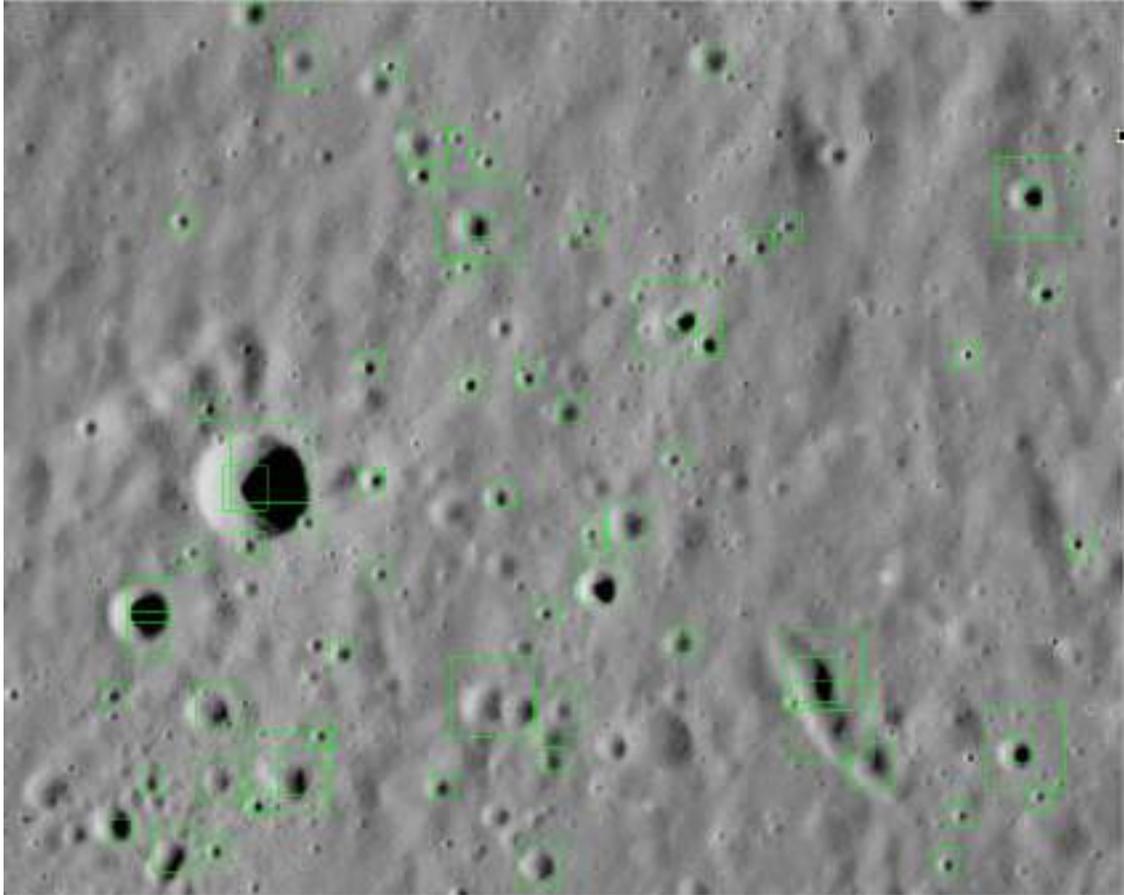


Figura 5.6: Detección de cráteres en la imagen 1 utilizando “*Adaptive Boosting*” con un filtrado rápido.

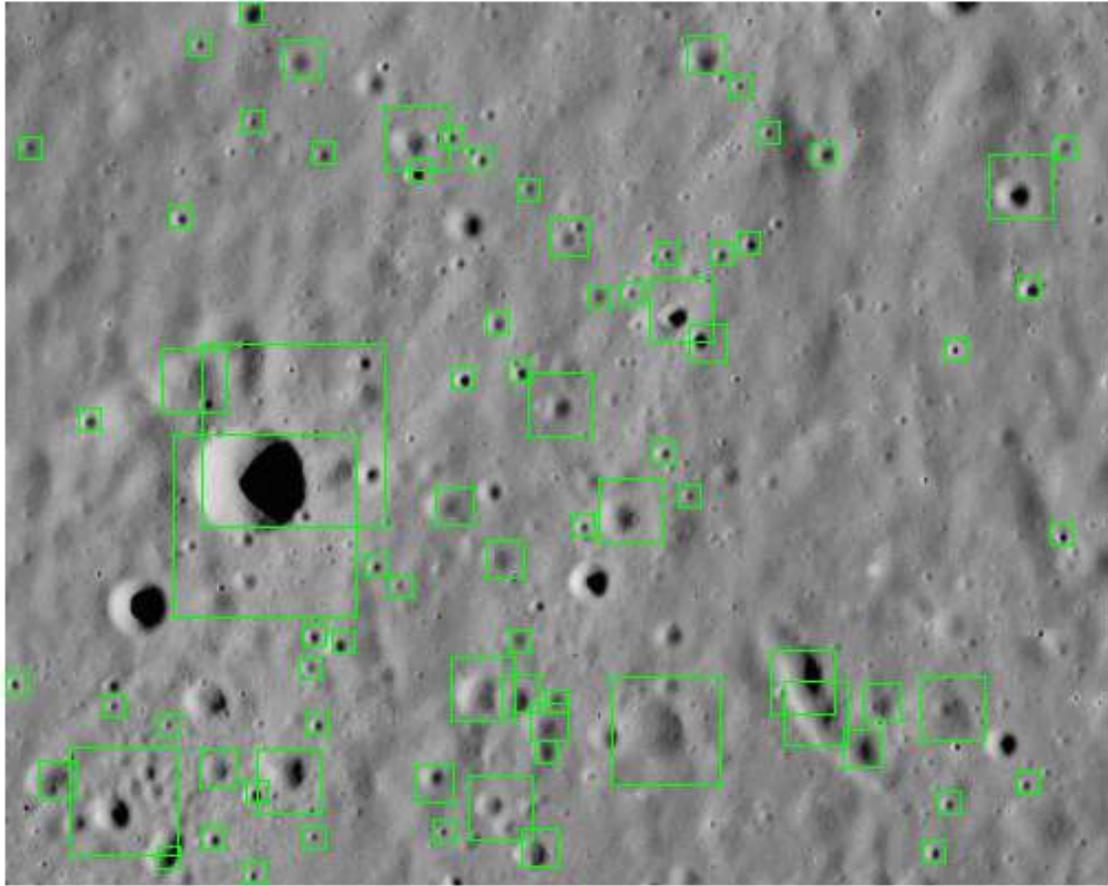


Figura 5.7: Detección de cráteres en la imagen 1 utilizando “*Adaptive Boosting*” con el filtrado automático propuesto.

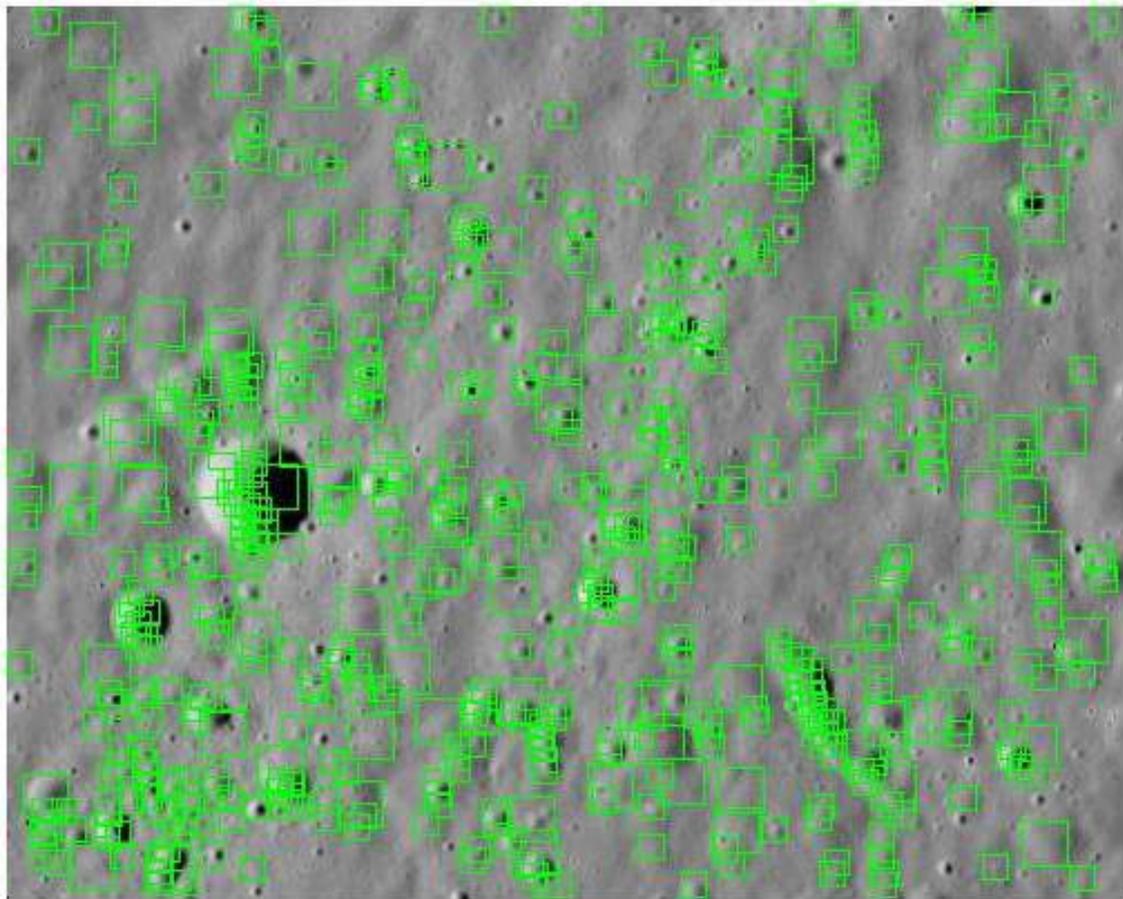


Figura 5.8: Detección de cráteres en la imagen 1 utilizando SVM con un filtrado rápido.

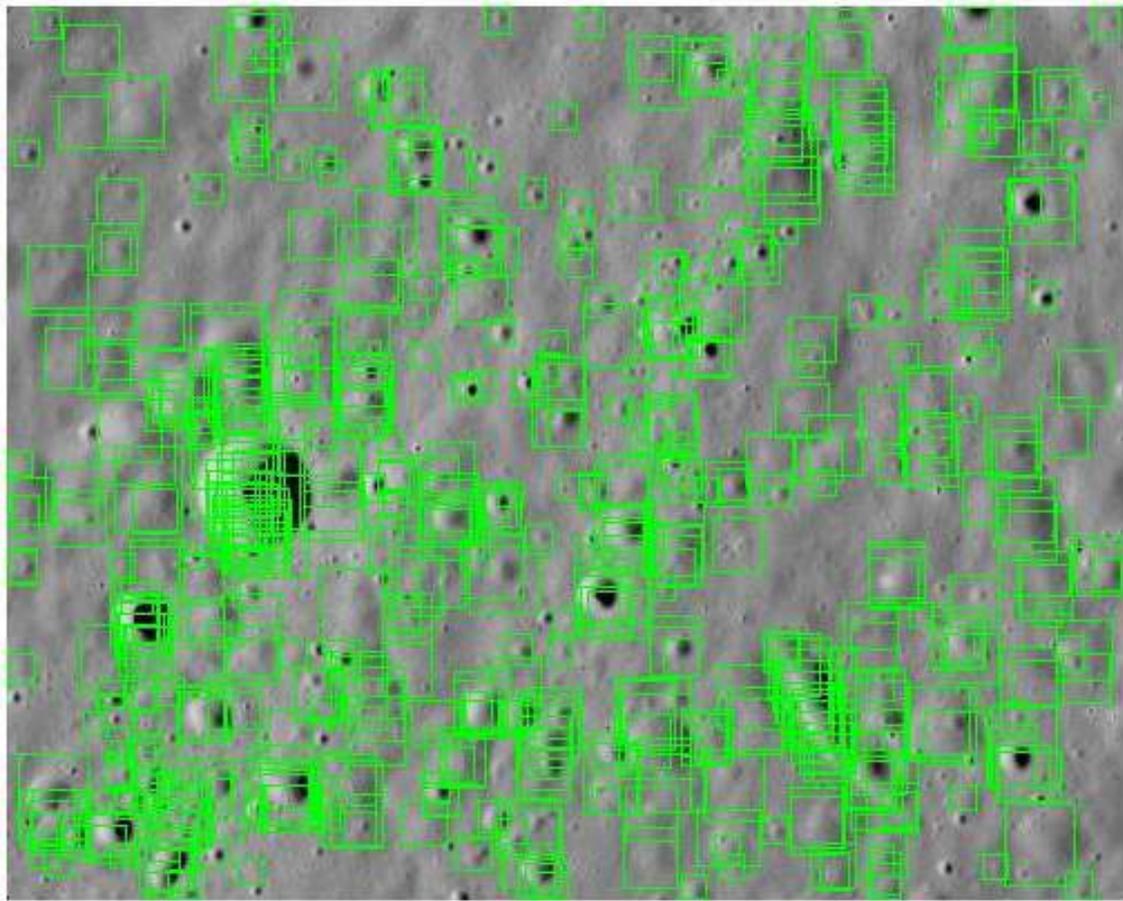


Figura 5.9: Detección de cráteres en la imagen 1 utilizando SVM con el filtrado automático propuesto

Utilizando los resultados obtenidos de la primera imagen, se realizó una tabla comparativa de los cráteres detectados, así como el aquellos que fueron detecciones erróneas. Se contabilizaron los cráteres detectados en cada imagen, así como las detecciones erróneas.

Característica	Cráteres Detectados	Falsos Positivos
“AdaBoost” Filtrado 1	56	6
“AdaBoost” Filtrado 2	72	12
SVM1 Filtrado 1	98	más de 60
SVM1 Filtrado 2	102	más de 60
Detección manual	126	0

Tabla 5.13: Comparación de los cráteres detectados y los cráteres etiquetados manualmente para la imagen 1. El filtrado 1 corresponde al filtrado “rápido” y el 2 es el filtrado automático propuesto.

La segunda imagen utilizada, consiste en una región del mar de la humedad, al igual que las primeras pruebas, las dos primeras imágenes muestran el mar de la humedad con su respectivo etiquetado manual de los cráteres. De igual manera se estableció el mismo tamaño mínimo de 6x6 píxeles.

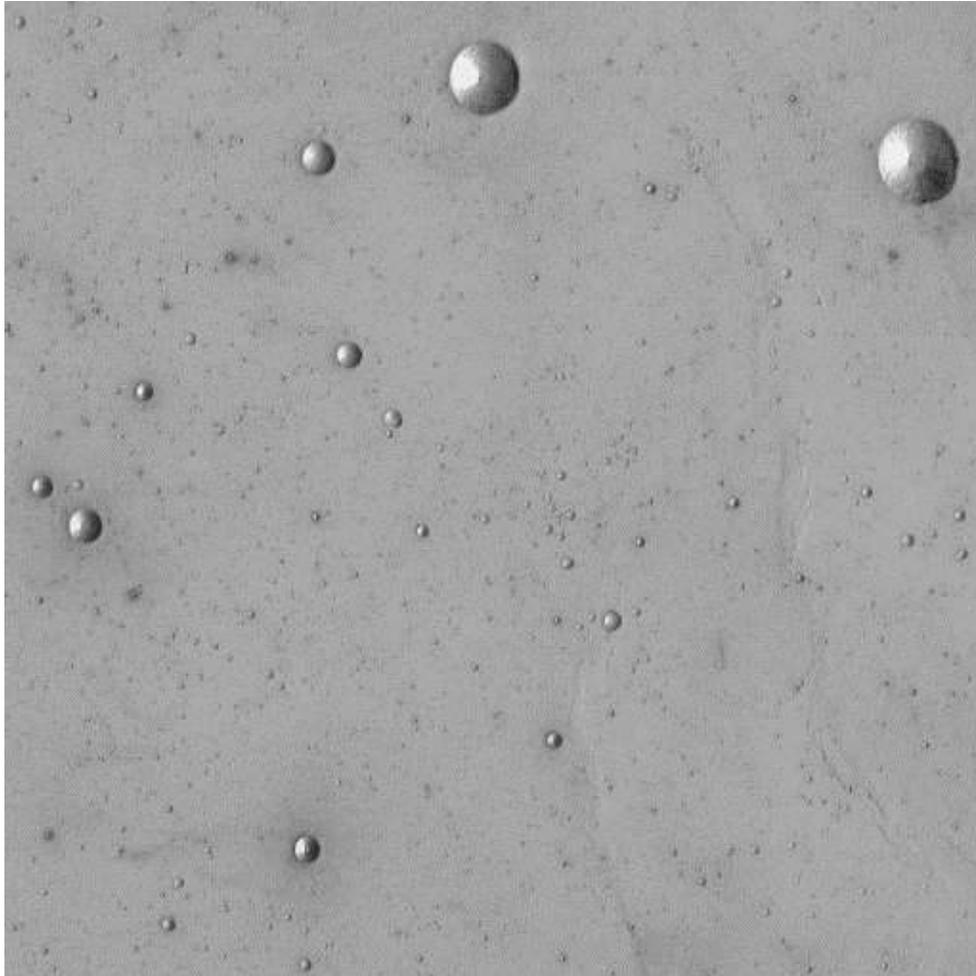


Figura 5.10: Imagen del Mar de la Humedad (Imagen de prueba 2).

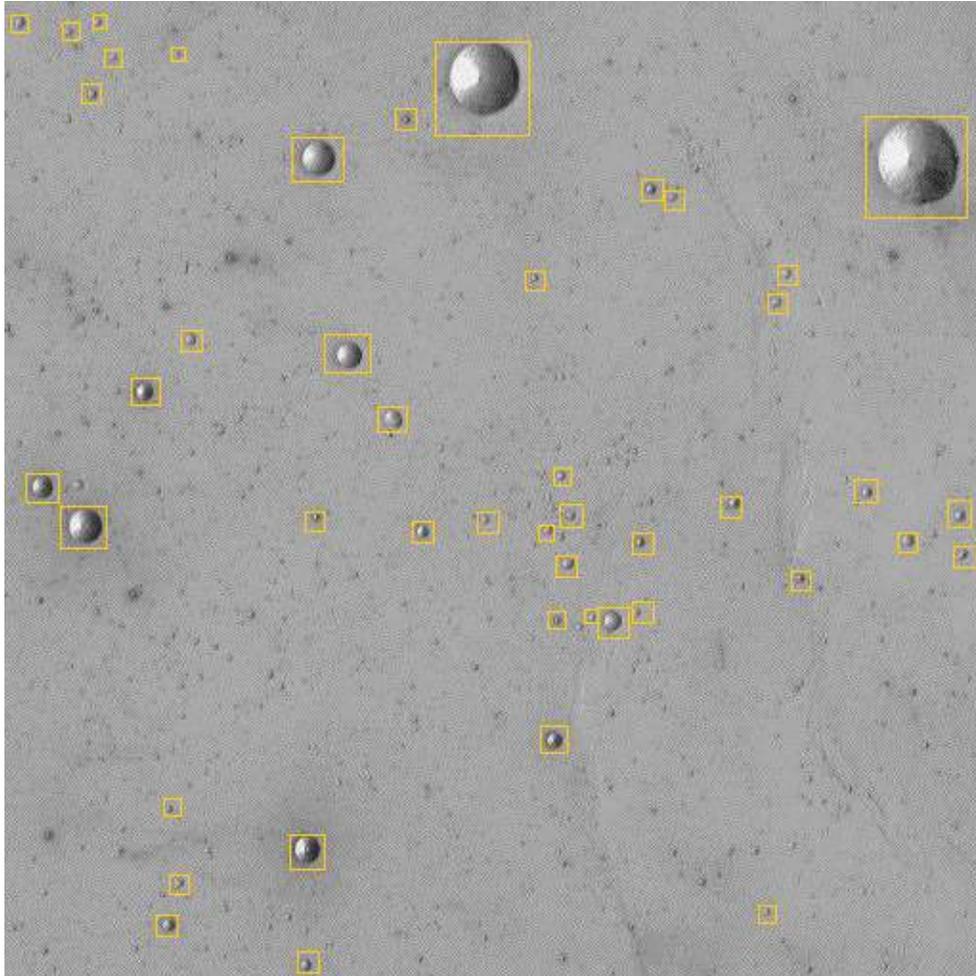


Figura 5.11: Etiquetado manual de cráteres presentes en la imagen de prueba 2.

En las siguiente imágenes podemos comparar los distintos filtros con ambos clasificadores, resultando de igual manera que el algoritmo de “*Adaptive Boosting*” tuvo un mejor desempeño general, cabe señalar que esto puede ocurrir debido a que las características fueron elegidos gracias al algoritmo de “*Adaptive Boosting*”; lo cual puede no ser la mejor opción para conformar el vector de características utilizando en los SVM.

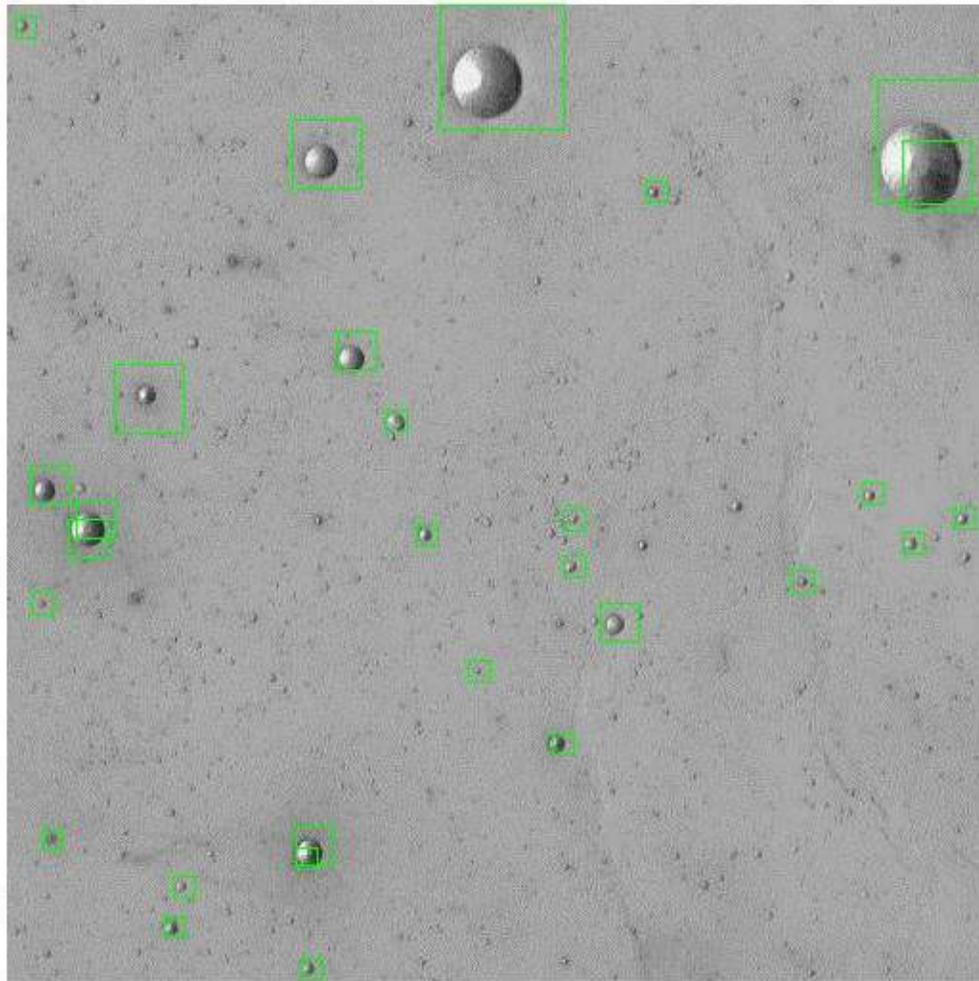


Figura 5.12: Detección de cráteres en la imagen 2 utilizando “*Adaptive Boosting*” con un filtrado rápido.

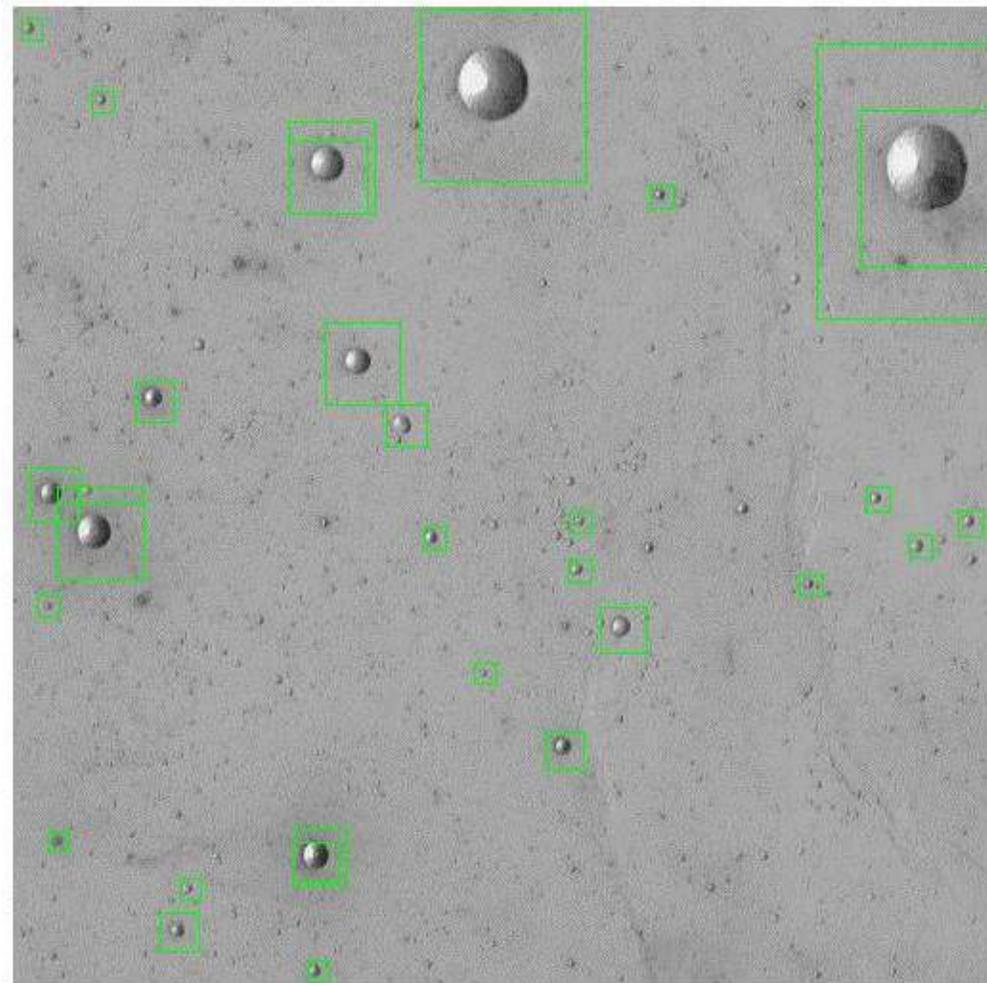


Figura 5.13: Detección de cráteres en la imagen 2 utilizando “*Adaptive Boosting*” con el filtrado automático propuesto.

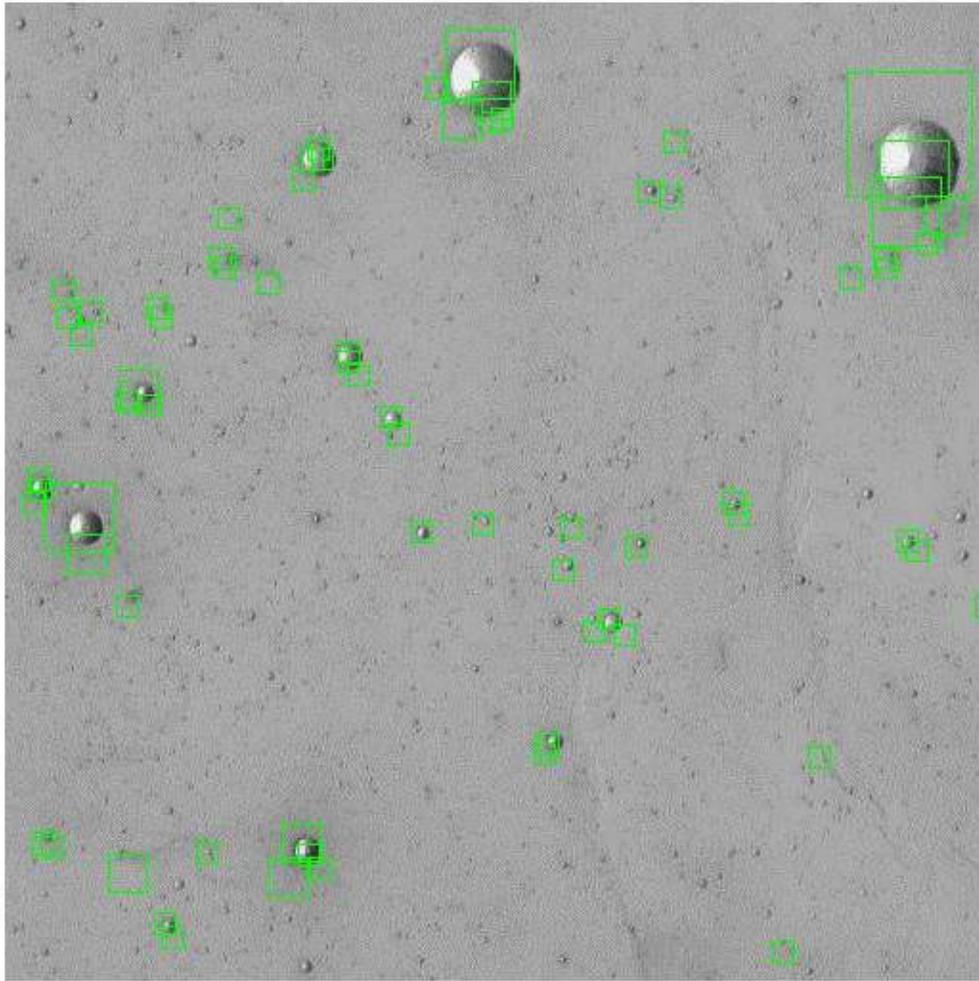


Figura 5.14: Detección de cráteres en la imagen 2 utilizando SVM con un filtrado rápido.

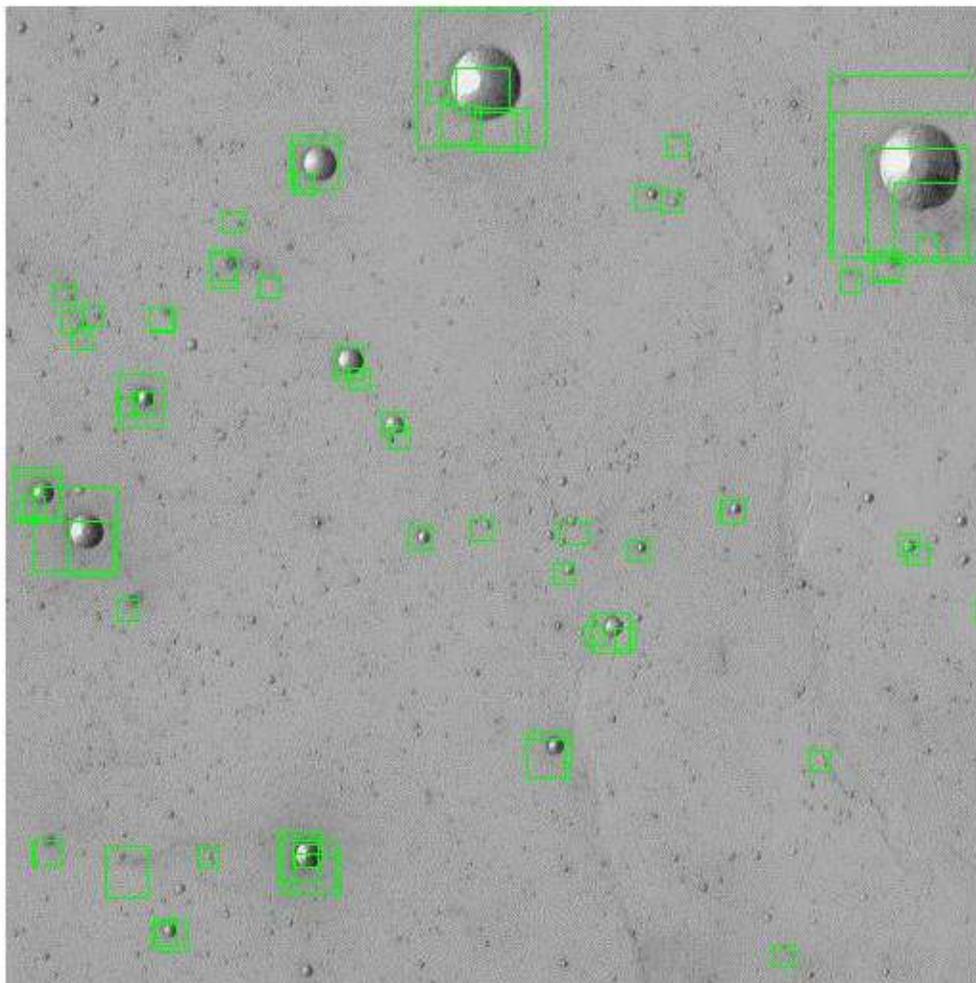


Figura 5.15: Detección de cráteres en la imagen 2 utilizando SVM con el filtrado automático propuesto.

Al igual que en la tabla de cráteres detectados que obtuvimos de la primera imagen, de igual manera se realizó una tabla comparativa de los cráteres detectados, así como el aquellos que fueron detecciones erróneas de los distintos modelos aplicados en la segunda imagen. Se contabilizaron los cráteres detectados en cada imagen, así como las detecciones erróneas.

Característica	Cráteres Detectados	Falsos Positivos
“AdaBoost” Filtrado 1	21	3
“AdaBoost” Filtrado 2	24	3
SVM1 Filtrado 1	27	35
SVM1 Filtrado 2	29	46
Detección manual	43	0

Tabla 5.14: Comparación de los cráteres detectados y los cráteres etiquetados manualmente para la imagen 2. El filtrado 1 corresponde al filtrado “rápido” y el 2 es el filtrado automático propuesto.

# Capítulo 6

## Conclusiones

Se concluye esta tesis con un resumen de los logros alcanzados en la investigación, así como los futuros trabajos que parten de la investigación y el trabajo logrado a lo largo de la elaboración de esta tesis.

### 6.1. Conclusiones

En este trabajo se propuso el uso de una metodología para la detección de cráteres lunares, en donde se utilizaron plantillas sugeridas además de las propuestas en trabajos previos. De igual manera se utilizó una propuesta de filtrado para la eliminación de los falsos positivos que dejó el primer proceso. Esta área es de mucho interés, ya que está en constante estudio la elección de características que se utilizaran en los clasificadores como lo son el *SVM* y el “*Adaptive Boosting*”. Como pudimos observar el uso de características Haar especializadas para extraer características de un objeto, presenta mejor rendimiento que las característica Haar generales que podemos observar en [42, 23], permitiendo que con un ajuste de umbral se detecte una mayor cantidad de grupos de píxeles que representan un cráter. Todo ello sumado a un filtrado especializado para este problema nos permite alcanzar un rendimiento óptimo. Entre las ventajas que se pudieron observar al momento de comparar los resultados, es que las características Haar propuestas englo-

ban mejor las características que representan a los cráteres, permitiendo que se diferencien del resto de las formaciones topográficas que encontramos en la luna. Sin embargo, debido a que las características Haar utilizadas para el entrenamiento dependen de la fuente de donde se extrae la información, en este caso de las imágenes de entrenamiento, esto tiene sus desventajas y ventajas ya que permite tener un clasificador óptimo para la detección de cráteres de zonas similares a las que se utilizaron para en entrenamiento del modelo, pero impiden generalizar el modelo; teniendo problemas en imágenes cuyas condiciones de luminosidad o resolución son distintas a las utilizadas por el entrenamiento. Lo anterior aparentemente se solucionó construyendo el sistema propuesto obteniendo una detección más limpia de los cráteres, sin embargo algunos parámetros no fueron automáticos e hizo ajustarlos, lo cual es un punto a seguir en los trabajos futuros. Finalmente, se probó que en el problema de detectar cráteres es importante conocer el objeto, ya que eso nos permite crear plantillas Haar especializadas, así como métodos de filtrado específicamente para las condiciones que presenta el entorno donde se encuentra presente el objeto a identificar. Con la metodología propuesta se obtuvo un incremento en la tasa de detección eso nos puede llevar a pensar que se deben introducir distintas metodologías, o incluir clasificadores híbridos para el reconocimiento de cráteres en superficies de astros y satélites.

Las contribuciones de este trabajo incluyen algoritmos para el entrenamiento de clasificadores utilizando las plantillas Haar propuestas, las cuales incrementaron la tasa de detección. Igualmente se contribuye con un método global de detección de cráteres, lo cual es de gran ayuda para la obtención de la distribución tamaño-frecuencia de los cráteres detectados. Otra de las contribuciones es un artículo de investigación publicado en el congreso no. 11th de (CCE) *Computing Science and Automatic Control* en donde se presentaron los avances preliminares de esta tesis. Por último, derivado de la obtención de imágenes utilizadas para el entrenamiento de los modelos, tenemos una base de datos de imágenes de cráteres lunares, (con sus respectivas muestras negativas), la cual permitirá el desarrollo y evaluación de futuras investigaciones en el área.

## 6.2. Trabajo Futuro

Siguiendo la línea del procesamiento de imágenes planetarias, así como la detección de características para el estudio del astro, los trabajos a futuro incluyen la construcción de un método que a partir de los cráteres detectados, calcule la distribución frecuencia-tamaño para de esa manera obtener las comparaciones con respecto a los demás astros. Esta nueva propuesta a futuro permitirá obtener una herramienta muy útil para el estudio de la edad de ciertas secciones de la luna o en su defecto otros astros.

En la metodología para aplicar el filtrado surge la pregunta ¿Qué filtros deberían ser más óptimos para evitar que se descarten muestras positivas? , lo cual es una problemática que se debe estudiar.

En el proceso de filtrado final igualmente encontramos otra pregunta a resolver ¿Cómo seleccionar y/o eliminar las muestras detectadas varias veces con distintos tamaños de subventana?. Esta problemática la encontramos al tener varias detecciones de un mismo cráter y se determinó utilizar un método donde se seleccionan todas las detecciones pintando una nueva subventana que abarcara dichas detecciones. Preguntarse como optimizar este proceso es una interrogante que se encontró ya que en el proceso se eliminaban cráteres individuales detectados.

Hablando de manera general de la detección de cráteres lunares, futuros trabajos estarían enfocados a la mejora del proceso, así como el diseño de mejores características, como el procesamiento de la imagen de manera que exista menos dependencia de los modelos elegidos para entrenar y sirvan solo de referencia en todo el proceso. Cambiar ese enfoque de aprendizaje guiado a uno semi-guiado donde el preprocesamiento haga más generales las características permitiría aumentar el desempeño del clasificador, posiblemente solucione esa dependencia, cosa que se verá en futuros trabajos.

Considerando de igual manera la desventaja del entrenamiento con “*Adaptive Boosting*” debido al tiempo de entrenamiento entre partición y partición, de igual manera se explora la idea de hacer un entrenamiento más optimizado paralelizando el proceso, para de esa manera hacer varios procesos de refinamiento y elección de características que son necesarios igualmente en otros clasificadores como los son los *SVM*.

De esto se puede concluir que aún hay mucho camino que recorrer, al igual que se exploró una nueva solución y se buscaron nuevas plantillas y métodos, es necesario incrementar el rendimiento del clasificador y de igual manera realizarlo de forma más general. Sin duda surgen muchas interrogantes y aspectos no triviales que nos hacen abordar el problema desde distintos enfoques para cubrir por completo la problemática, este problema tiene mucho campo de estudio en distintas áreas.

# Bibliografía

- [1] <http://wms.lroc.asu.edu/apollo/browse?camera=M&mission=15>, Octubre 2014.
- [2] Lourenço Bandeira, José Saraiva, and Pedro Pina. Impact Crater Recognition on Mars Based on a Probability Volume Created by Template Matching. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4008–4015, dec 2007.
- [3] Teresa Barata, E Ivo Alves, José Saraiva, and Pedro Pina. Automatic Recognition of Impact Craters on the Surface of Mars. *Springer-Verlag Berlin*, pages 489–496, 2004.
- [4] J Canny. A Computational Approach to Edge Detection. *IEEE-Trans Pattern Analysis and Machine Intelligence*, (10(6)), 1986.
- [5] César Iván Cobos-May. *Algoritmos para diseñar plantillas de convolución que determinan características de tipo Haar*. PhD thesis, Universidad de Yucatán, 2013.
- [6] Wei Ding, Tomasz F Stepinski, Yang Mu, Lourenco Bandeira, Ricardo Ricardo, Youxi Wu, Zhenyu Lu, Tianyu Cao, and Xindong Wu. Sub-Kilometer Crater Discovery with Boosting and Transfer Learning. V:1–25, 2010.
- [7] Wei Ding, Tomasz F Stepinski, Yang Mu, Lourenco Bandeira, Ricardo Ricardo, Youxi Wu, Zhenyu Lu, Tianyu Cao, and Xindong Wu. Sub-Kilometer Crater Discovery with Boosting and Transfer Learning. V:1–25, 2010.
- [8] R. Duda and P. Hart. Use of the Hough Transform to Detect Lines and Curves in Pictures. *Comunnications of the Association for Computing Machinery*, (15), 1972.
- [9] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, may 1999.

- [10] Bevan M French. Traces of catastrophe: A handbook of shock-metamorphic effects in terrestrial meteorite impact structures. *Technical Report, LPI-Contrib-954*, 1, 1998.
- [11] M. Frigo and Johnson S. G. FFTW: An adaptive software architecture for the FFT. *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 3:1381–1384, 2007.
- [12] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. 2007.
- [13] I. Guyon, S. Gunn, M. Nikraves, and L. A. Zadeh. *Feature Extraction Foundations and Applications*. Springer, 2006.
- [14] R.M. Hadad, A. De A Araujo, and P.P. Martins. Using the Hough transform to detect circular forms in satellite imagery. *Proceedings XIV Brazilian Symposium on Computer Graphics and Image Processing*, page 406, 2001.
- [15] Jiang He, Hutao Cui, and Feng Junhua. Edge Information Based Crater Detection and Matching for Lunar Exploration. *International Conference on Intelligent Control and Information Processing*, pages 302–307, 2010.
- [16] R. Honda, Y. Iijima, and O. Honishi. Mining of topographic feature from heterogeneous imagery and its applications to lunar craters. *Springer-Verlag, London Uk*, 2002.
- [17] Boris A Ivanov. Mars/moon cratering rate ratio estimates. In *Chronology and evolution of Mars*, pages 87–104. Springer, 2001.
- [18] Ameer Jain, Darshak Thakore, and Narendra Patel. Automatic crater detection on lunar surface. *International Journal of Innovative Research in Science, Engineering and Technology*, 2(5):1448–1453, 2013.
- [19] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [20] B. Leroy, G. G. Medioni, E. Johnson, and Matthies L. Crater Detection for autonomous landing on asteroids. *Image Vis. Comput.*, 19(11):787–792, September 2001.
- [21] Pierre Lison. *An introduction to machine learning*. 1996.
- [22] Lei Luo, Xin-yuan Wang, Wei Ji, and Chao Li. Automated detection of lunar craters based on Chang'E-1 CCD data. *2011 4th International Congress on Image and Signal Processing*, pages 883–887, oct 2011.

- [23] R. Martins, P. Pina, J.S. Marques, and M. Silveira. Crater Detection by a Boosting Approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1):127–131, jan 2009.
- [24] H Jay Melosh. Impact cratering: A geologic process. *Research supported by NASA. New York, Oxford University Press (Oxford Monographs on Geology and Geophysics, No. 11), 1989, 253 p., 1, 1989.*
- [25] Yang Mu, Wei Ding, Dacheng Tao, and T.F. Stepinski. Biologically inspired model for crater detection. *The 2011 International Joint Conference on Neural Networks*, pages 2487–2494, jul 2011.
- [26] Edward Murphy and Randy Bell. Teaching lunar stratigraphy and the nature of science. pages 34–40, 2013.
- [27] G Neukum. The Inner Solar System Impact Record: Lunar, Terrestrial-Planet, and Asteroid size-frequency data comparison. *LSPC XXV*.
- [28] G Neukum and B. A. Ivanov. The size-frequency distribution of impact craters in the inner solar system and implications for past and recent cratering rates in the Earth-Moon.
- [29] Gerald North. Observing the Moon The modern astronomers guide. *Cambridge University Press*, 200.
- [30] C. Papageorgiou, T. Oren, and Poggio. A general framework for object detection. In: Sixth International Conference on Computer Vision. pages 555–562, 1998.
- [31] C. Papageorgiou, T. Oren, and Poggio. A general framework for object detection. In: Sixth International Conference on Computer Vision. pages 555–562, 1998.
- [32] Ernesto Pérez. Caos en el sistema solar. pages 59–69, 1998.
- [33] G. Salamuniccar and Loncaric S. Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on MOLA data. *Advances in Space Research*, pages 6–19, 2007.
- [34] S Sangeetha and Radha N. A new framework for iris and fingerprint recognition using SVM classification and extrema learning machine based on score level fusion. pages 183–188, January, 2013.

- [35] Y Sawabe, T Matsunaga, and S Rokugawa. Automated detection and classification of lunar craters using multiple approaches. *Advances in Space Research*, 37(1):21–27, 2006.
- [36] J. Serra. Image Analysis and Mathematical Morphology. *Academic Press, New York*, 1982.
- [37] L. G. Shapiro and G. C. Stockman. Computer Vision. *Prentice Hall*, 2001.
- [38] S. Spitalnic. Test properties I: Sensitivity, especificity and predictive values. September 2004.
- [39] Tomasz Stepinski, Ricardo Vilalta, and Soumya Ghosh. Machine Learning Tools for Automatic Mapping of Martian Landforms. *IEEE Intelligent Systems*, 22(6):100–106, 2007.
- [40] G. Troglio, J. a. Benediktsson, G. Moser, S. B. Serpico, and J. Le Moigne. Automatic Extraction of Planetary Image Features. *2009 Third IEEE International Conference on Space Mission Challenges for Information Technology*, pages 211–215, jul 2009.
- [41] Hough Paul VC. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.
- [42] Paul Viola, Mitsubishi Electric, and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. 2001.
- [43] P.G. Wetzler, R. Honda, B. Enke, W.J. Merline, C.R. Chapman, and M.C. Burl. Learning to Detect Small Impact Craters. *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, pages 178–184, jan 2005.