



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS

**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS**

---

---

FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA  
MAESTRIA EN OPTIMIZACION Y COMPUTO APLICADO

## **VISION ARTIFICIAL Y MACHINE LEARNING**

### **MODELO DE DELIMITACIÓN DE TECHOS PARA EL USO DE PANELES FOTOVOLTAICOS BASADO EN INTELIGENCIA ARTIFICIAL**

Que para obtener el Grado de MAESTRO EN OPTIMIZACIÓN Y COMPUTO  
APLICADO

Presenta

**GEAN CARLO VITORINO MONJARAS**

Director de Tesis

OUTMANE OUBRAM

Co-Director

BASSAM ALI

Revisores:

LORENA DIAZ GONZALEZ, FEDERICO ALONSO PECINA, ROY LOPEZ



CUERNAVACA, MORELOS

JULIO ,2020



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



## FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a \_03 de Julio\_ del 2020.

**DR. AUGUSTO RENATO PÉREZ MAYO**  
**SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la *Maestría en: Optimización y Computo Aplicado*, del estudiante Gean Carlo Vitorino Monjaras, con matricula 10023099, con el título **MODELO DE DELIMITACIÓN DE TECHOS PARA EL USO DE PANELES FOTOVOLTAICOS BASADO EN INTELIGENCIA ARTIFICIAL** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
***Por una humanidad culta***  
*Una universidad de excelencia*

**Oubram Outmane**  
**Profesor- investigador de tiempo completo**  
**Facultad de Ciencias Químicas e Ingeniería**



UADY

FACULTAD DE INGENIERÍA

Mérida, Yucatán a 28 de agosto del 2020.

**DR. AUGUSTO RENATO PÉREZ MAYO**  
**SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la *Maestría en: Optimización y Computo Aplicado*, del estudiante ***Gean Carlo Vitorino Monjaras***, con matrícula ***10023099***, con el título ***MODELO DE DELIMITACIÓN DE TECHOS PARA EL USO DE PANELES FOTOVOLTAICOS BASADO EN INTELIGENCIA ARTIFICIAL*** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además, construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente

**Bassam Ali**  
**Profesor- investigador de tiempo completo**  
**Facultad de Ingeniería**  
**Universidad Autónoma de Yucatán**

Campus de Ciencias Exactas e Ingenierías  
Av. Industrias No Contaminantes por Anillo Periférico Norte S/N. Apartado Postal núm. 150 Cordemex  
Teléfono: (999) 930 05 50, 930 05 66 | Mérida, Yucatán, México | [www.uady.mx](http://www.uady.mx)



Cuernavaca, Morelos a 28 de agosto del 2020.

**DR. AUGUSTO RENATO PÉREZ MAYO**  
**SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la *Maestría en: Optimización y Computo Aplicado*, del estudiante ***Gean Carlo Vitorino Monjaras***, con matricula ***10023099***, con el título ***MODELO DE DELIMITACIÓN DE TECHOS PARA EL USO DE PANELES FOTOVOLTAICOS BASADO EN INTELIGENCIA ARTIFICIAL*** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además, construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
***Por una humanidad culta***  
*Una universidad de excelencia*

Dr. Lorena Díaz González  
Profesora investigadora de tiempo completo  
Centro de Investigación en Ciencias



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

**Sello electrónico**

LORENA DIAZ GONZALEZ | Fecha:2020-08-29 16:50:40 | Firmante  
GeOYnG+c+4zkFFajvNEN8VWUyITJXbBtAlhE9mNIUAAF4qMoZgg4j6g56MQ3MkuwA13F0fGm0RvBKnw4udcvKT5Y0XLetbmfYw+q1YRM0ld0RQzskD8fEx0rzF3F18rJU+M8R  
0FMXurSZX5rN3qnzqzpjCi2APrK79Baw0nS9EvqGjKTZLT9f545fuxx5VlBmylThAvYznT/MFwrcf6CHbBqbrFBwk2eFYnkCb37Tl0/exXKvJtry120wMm3YxJdbmGkkt70Ka3nUvaZT  
1Zqjcl1JpJjSxW8esJvRkUxr0rvcnDqFn1A4oyVRcx8yo94LMJuROZrqRbDvcK+3Q==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o  
escaneando el código QR ingresando la siguiente clave:



dIZJME

<https://efirma.uaem.mx/noRepudio/GwYZAf3Inqwg9UXSIFEzxeNSuWp7xTF>



Una universidad de excelencia

RECTORÍA  
2017-2023



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS



## FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a 28 de agosto del 2020.

**DR. AUGUSTO RENATO PÉREZ MAYO**  
**SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la *Maestría en: Optimización y Computo Aplicado*, del estudiante **Gean Carlo Vitorino Monjaras**, con matricula **10023099**, con el título **MODELO DE DELIMITACIÓN DE TECHOS PARA EL USO DE PANELES FOTOVOLTAICOS BASADO EN INTELIGENCIA ARTIFICIAL** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además, construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
**Por una humanidad culta**  
*Una universidad de excelencia*

**Federico Alonso Pecina**  
**Profesor- investigador de tiempo completo**  
**Facultad de Contaduría, Administración e Informática**

Av. Universidad 1001 Chamilpa Cuernavaca Morelos México C.P. 62209, Edificio 19  
Tel. (777) 329 7917, Ext. 3038, 3039/ posgrado.fcaei@uaem.mx



*Una universidad de excelencia*

RECTORÍA  
2017-2023



UNIVERSIDAD AUTÓNOMA DEL  
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

#### Sello electrónico

FEDERICO ALONSO PECINA | Fecha:2020-08-28 19:23:52 | Firmante

tQH5oqGQ0hATrkkxAaoVWJlaYOloOX8QJLPq9mJBLUz8Ds396ubbPrt4aTRgorLxaQWnw4+J/pt1m2uICSKd0JaOUi2I//Lu3biVdqBvCRqJ/X8popNsDYeCpeex0JOT0RVpp2Zdrx  
cYzaziXz0trpz9k+oUEshite7gSAlumz6VSM04w9cbI23h8bR116IHmZWecy+FBE/oWHqpOZmz0JQh9wV/XzjhBYk5TtoERx40yRcLUAFcMUT4JUdJl/PL0yKaC0JVh3KLqIQ+6tt9YR  
dlik4+G4FLzmaLFS00XSmDALZgkEz91r3WKgkVWgRx2Mqwx+bEagJ2Co4D0Oy89w==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o  
escaneando el código QR ingresando la siguiente clave:



jbr4gK

<https://efirma.uaem.mx/noRepudio/d7Gkzm0cRwNObP9eVDPyVv3nAgaDVh1E>

UA  
EM

Una universidad de excelencia

RECTORÍA  
2017-2023



Cuernavaca, Morelos a 30 de agosto del 2020.

**DR. AUGUSTO RENATO PÉREZ MAYO**  
**SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI**  
**PRESENTE**

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la *Maestría en: Optimización y Computo Aplicado*, del estudiante ***Gean Carlo Vitorino Monjaras***, con matrícula ***10023099***, con el título ***MODELO DE DELIMITACIÓN DE TECHOS PARA EL USO DE PANELES FOTOVOLTAICOS BASADO EN INTELIGENCIA ARTIFICIAL*** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además, construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente  
***Por una humanidad culta***  
*Una universidad de excelencia*

**Dr. Roy López Sesenes**  
Profesor Investigador de Tiempo Completo Titular A  
Facultad de Ciencias Químicas e Ingeniería-UAEM  
e-mail: [rlopez@uaem.mx](mailto:rlopez@uaem.mx), ext. 2089





Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

**Sello electrónico**

ROY LOPEZ SESENES | Fecha:2020-08-30 21:44:18 | Firmante  
EB8tv5i0umbu+bYcPfdPU8Qrq8/CkvzphEnK2ZHE7YPdelfUSsj0plCus0dilmIKfIM/qdSGXRuggerB5HOuSTzeZUzW03Pc6ExCn58BsknyDgVtVplQoW7G+h4L1DBafUPK0CbnMt  
MlnwteMErMpYiO71JX22ylaEkJkLcV4llv+cXln0XkpuXVZTU/Z8EvNHZ+Tz6A1zjGd2rDvJmcC9lj/Dh086O7BXgFHg2fmyMw0maOn7o8bgwbiD8i1AP+IkOrX7rOrwHlzi1PUIEtOijp  
STLM+jJA9Gdy6TFUhterGDLy3Qm52Ggpm7KTZYRYu4yqnWknauXAIBjrGV4Dw==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o  
escaneando el código QR ingresando la siguiente clave:



XOzbCq

<https://efirma.uaem.mx/noRepudio/CXdfVkbCo5n7UOTOs7sgl7kj8agwkF5>



## **DEDICATORIA**

A mi pareja, padres y hermano por siempre darme el apoyo que necesito,  
Y a todos los investigadores y autores que me inspiraron,  
Por el deber de mejorar el mundo.

## **AGRADECIMIENTOS**

A mi Familia por la paciencia y el apoyo.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) y Sistema Nacional de Investigadores (SNI), por la beca de nacional de posgrado otorgada para la realización de mis estudios de maestría.

A la secretaria de Investigación y Posgrado de la facultad de contaduría, Administración e Informática (FCAeI) por admitirme en la segunda generación de la Maestría de Optimización y Computo Aplicado.

Al Dr. Outmane Oubram por aceptarme como tesista, su disposición, tiempo, y el aprendizaje tan basto y grato que con gusto me compartió, y gran agradecimiento por introducirme y darme las bases al mundo del tratamiento de imágenes digitales.

Al Dr. Bassam Ali por aceptarme como tesista, su disposición y tiempo para el aprendizaje que me otorgo a lo largo de esta travesía, además de creer en mi desde la licenciatura y alentarme a ir más lejos con mis objetivos.

A mi Comité tutorial, Dra. Lorena Díaz, Dr. Federico Alonso, Dr. Roy Lopez, Dr. Rachid Marzoug; quienes dieron seguimiento y asesoría a mis avances durante todo el tiempo en el que este trabajo fue realizado.

# Contenido

Índice de Tablas.....	14
Índice de Figuras .....	15
Capítulo 1 .....	18
INTRODUCCIÓN .....	18
Estado del arte y Justificación .....	19
Objetivos .....	23
Metodología .....	23
Estructura de la Tesis .....	24
Capitulo II .....	26
Algoritmo de Segmentación .....	26
Fundamentos .....	26
Etapas procesamiento de imágenes .....	33
Metodología: Algoritmo de segmentación de la Imagen Digital .....	37
NVDI .....	41
Filtro Gaussiano.....	44
Método Otzu .....	47
Distancia Euclidean.....	49
Erosión.....	51
Apertura .....	52
Aplicación del Algoritmo Segmentado.....	53
Capitulo III .....	58
Clasificador con Aprendizaje de Maquina.....	58
Fundamentos .....	58
Metodología de construcción del modelo ML .....	64
Data-set .....	66
Métricas.....	67
Features.....	73
Análisis y experimentación de modelos <i>ml</i> .....	84
SVM Polinomial .....	94
Capitulo IV .....	100
Aplicación.....	100
Algoritmo integrado.....	100

Aplicación .....	101
Prueba Experimental.....	104
Capítulo V.....	108
Conclusiones .....	108
Conclusiones.....	108
Objetivos logrados.....	109
Trabajos a Futuro .....	109
Referencias.....	110

## Índice de Tablas

Tabla 1	Características de imágenes digitales	33
Tabla 2	Índices sobre las clases de vegetación	37
Tabla 3	Mascara del Filtro Gaussiano	39
Tabla 4	Experimentos de la curva ROC	67
Tabla 5	vecindad $f(x, y)$ , al centro se sitúa el pixel con posición $(x, y)$	69
Tabla 6	Resultado de la relación entre pixeles	72
Tabla 7	Matriz Simétrica de co-ocurrencia	72
Tabla 8	Matriz de probabilidad	72

## Índice de Figuras

Figura 1	Flujo Metodológico	16
Figura 2	Color sensible del espectro humano	19
Figura 3	Inhibición Lateral	19
Figura 4	Mapa de Bitz	20
Figura 5	Representación Matricial Modelo RGB	21
Figura 6	Ejemplo de transformación de Calor	21
Figura 7	Tipos de Imágenes Digitales	22
Figura 8	Muestreo de una imagen RGB	23
Figura 9	Proceso de captura de una imagen digital	24
Figura 10	Ejemplos de resolución espacial	25
Figura 11	Metodologia del PDI	25
Figura 12	Vecindad de una imagen Digital	26
Figura 13	Métodos para la extracción de características	27
Figura 14	Aplicación de una imagen macro	29
Figura 15	Algoritmo de segmentación por métodos	30
Figura 16	Muestra 1 Imagen Satelital RGB	32
Figura 17	Imagen satelital en escala de grises	33
Figura 18	Algoritmo modificado del NVDI	34
Figura 19	Imagen Satelital solo de Vegetación	35
Figura 20	Imagen Satelital con NVDI aplicado	35
Figura 21	Ruido Gaussiano	36
Figura 22	Mascar de Filtro Espacial	36
Figura 23	Imagen con filtro gaussiano aplicado	37
Figura 24	Histograma de imagen con ruido	38
Figura 25	Histograma de con el filtro gaussiano aplicado	38
Figura 26	Histograma Método Otzu	40
Figura 27	Imagen Escala de grises del Método Otzu	41
Figura 28	Ejemplo de Transformada de Distancia Euclidean	42
Figura 29	Imagen Satelital con la Distancia Euclidean Establecida	43
Figura 30	Ejemplo de Operación Morfológica Erosión	44
Figura 31	Imagen Satelital con Erosión Aplicada	45
Figura 32	Ejemplo del Método Apertura	45
Figura 33	Imagen segmentada	46
Figura 34	Imagen Satelital Caso 1	46
Figura 35	Imagen Resultante Caso 1 Segmentada	47
Figura 36	Imagen Satelital Caso 2	47
Figura 37	Imagen Resultante Caso 2 Segmentada	48
Figura 38	Imagen Satelital Caso 3	49
Figura 39	Imagen Resultante Caso 3 Segmentada	49

Figura 40	Imagen Satelital Caso 4	50
Figura 41	Imagen Resultante Caso 4 Segmentada	52
Figura 42	Distintos campos y aplicaciones del AI, ML y DL	53
Figura 43	Estructura de ML	54
Figura 44	Esquema General del Modelo ML	55
Figura 45	Ejemplo de Modelo de Clasificación de ML	55
Figura 46	Ejemplo simple de Modelo de Regresión lineal	56
Figura 47	Metodología Construcción de ML	57
Figura 48	Determinación de features para los modelos ML	58
Figura 49	Determinación de los hyper-parámetro en los modelos de ml	60
Figura 50	Muestra Conjunto de Datos	60
Figura 51	Method K-Fold Cross Validation	61
Figura 52	Funciona K-Fold	62
Figura 53	Matriz de Confusión	64
Figura 54	Matriz de Confusión Métricas	64
Figura 55	Precisión y exactitud demostración	65
Figura 56	Ejemplo de la curva l ROC	68
Figura 57	Imagen de ventana móvil de 5 x 5	69
Figura 58	Pixel y las direcciones que toma	69
Figura 59	Muestra de matriz de co-ocurrencia de 4 pixeles	74
Figura 60	Ejemplo de cuadro delimitador	74
Figura 61	Ejemplo de Centroide	75
Figura 62	Ejemplo de Área Convexa	77
Figura 63	Curva de Error Regresión Logística	78
Figura 64	Matriz de confusión de K fold de regresión logística	79
Figura 65	Métricas de regresión logística	79
Figura 66	Modelo ROC Regresión Logística	79
Figura 67	Matriz de confusión DGA	80
Figura 68	Métricas de DGA	81
Figura 69	Modelo ROC de DGA	81
Figura 70	Red Neuronal	83
Figura 71	Matriz de Confusión	83
Figura 72	Métricas ANN	83
Figura 73	Curvas del Modelo ROC	83
Figura 74	Matriz de Confusión SVM	85
Figura 75	Métricas de SVM	85
Figura 76	Modelo ROC de SVM	85
Figura 77	Cuadro comparativo de los modelos experimentados	86
Figura 78	Separación de clases de SVM Polinomial	87
Figura 79	Vectores de soporte ejemplo	87
Figura 80	Ejemplificación de separación de clases	88



Figura 81	Vectores de distancia	88
Figura 82	Descripción matemática de vectores de soporte	88
Figura 83	Matriz de confusión SVM Polinomial	91
Figura 84	Métricas SVM Polinomial	91
Figura 85	Experimentos Curvas ROC SVM polinomial	91
Figura 86	Diagrama de flujo Algoritmo Integrado	93
Figura 87	Muestra 1 Aplicación del modelo algoritmo integrado	94
Figura 88	Muestra 2 Aplicación del modelo algoritmo integrado	95
Figura 89	Muestra 3 Aplicación del modelo algoritmo integrado	95
Figura 90	Muestra 4 Aplicación del modelo algoritmo integrado	96
Figura 91	Muestra 5 Aplicación del modelo algoritmo integrado	96
Figura 92	Miscelánea Facultad de ingeniería	97
Figura 93	Facultad aplicado con el algoritmo integrado	98
Figura 94	Levantamiento de medidas	98
Figura 95	Croquis del Edificio a evaluar	99
Figura 96	Imagen original de edificio a evaluar	99
Figura 97	A) Imagen Segmentada, B) Imagen Clasificada	100

# Capítulo 1

## INTRODUCCIÓN

El consumo de energía eléctrica en las ciudades de todo el mundo ha estado aumentando año con año, y seguirá aumentando (CEPAL, 2018), y el sistema para satisfacer esta demanda están afectando nuestro ecosistema global. Con las nuevas tecnológicas en las ciencias de la computación se puede subsanar este problema con una propuesta de solución sustentable y dinámica. La clave de ello es ver el potencial de energías renovables que se encuentran en las ciudades, buscando la sustituir o cambiar toda la producción eléctrica de una ciudad de energías fósiles hacia energías renovables.

En efecto, las metodologías utilizadas para las planificaciones energéticas de una ciudad, se encuentran diversificadas e ineficientes, además de tener un desarrollo ad hoc, existen diferentes factores que causan este tipo de metodologías, la falta de la calidad de toma de datos (Gean et al, 2016), la falta de propuestas de modelos de datos. Otro punto crítico es que ninguna metodología propuesta se basa en las energías renovables como su apalancamiento en la producción energética, por ello no se puede visualizar un metodología estandarizada y sustentable.

Po otro lado, en los tiempos actuales donde las ciencias computacionales están en su auge, la inteligencia artificial (AI) está comenzando ser un pilar importante para la creación de herramientas que solucionen los principales problemas de nuestros tiempos. De igual manera la tecnología nos ha dado la forma de representar una ciudad en una imagen, ya sea con satélites, aviones o drones, tener la imagen de una ciudad con una resolución de calidad nos permite hacer una representación de toda una ciudad en datos, con ellos se puede obtener una vasta información para poder interpretar con el enfoque que uno desee. Con estos dos campos mencionados de las ciencias computaciones nos permite crear algoritmos, como herramientas robustas para clasificar o predecir. Ello es de mucha utilidad para ver el potencial de energías renovable dentro de una ciudad a través de estos datos.

La presente tesis propone un modelo de IA para ver el potencial de energía solar en los techos de una ciudad, gracias ello poder cuantificar que porcentaje de la demanda energética se puede abastecer con paneles fotovoltaicos, en vías de una ciudad sustentable.

En efecto el modelo del presente proyecto de investigación, se realizó gracias a diferentes campos de las ciencias de computación los cuales son, el desarrollo de un algoritmo clarificador de tratamiento de imágenes digitales, para descartar todo lo que no sean techos.

El aprendizaje de máquina, se entrenó un modelo de máquinas de soporte vectoriales con feature extraídos de la imagen satelital para la clasificación de techos factibles para la instalación de paneles fotovoltaicos, por último, se hizo un post-procesamiento de la imagen, se integró a un sistema de información geográfica, para poder hacer la cuantificación de los techos para la producción de los paneles fotovoltaicos, por usuario y por región.

El modelo propuesto podrá ayudar para la creación de herramientas futuras con el fin de una Metodología de planificación energética en vías de una ciudad sustentable, dándole a los tomadores de decisiones un instrumento más factible y certero, para planes de desarrollo sustentable.

## **Estado del arte y Justificación**

En la actualidad tenemos el objetivo nacional de la disminución o llegar a una producción casi nula de emisiones de dióxido de carbono en México para el 2050 acordado en un pacto mundial COP21 (Di Pietro, 2019). Pero no existe planes concretos para la transformación de las ciudades a ciudades sustentable, para la disminución de contaminación en este sector, siendo más específicos, en México no hay iniciativas para proyectos de planeación sustentables de transformación, y en sector de energía limpia solar tampoco (Jose C., 2016). Poniendo más detalle, en general hay pocas herramientas que funcionen de manera general que ayuden a cuantificar el potencial de energías solar en una ciudad.

Por otro lado, existen diferentes proyectos alrededor del Mundo que han propuestos herramientas en vías de desarrollo que muestran el potencial de energía solar de una ciudad, estos proyectos se asemejan en resultado y con algunas técnicas potenciales, entre los principales podemos nombrar:

- Mapwell System, fue desarrollado por el equipo del Instituto Tecnológico de Massachusetts (MIT) y la agencia de software MoDe Estudio, es una plataforma abierta de evaluación solar, que permite ver el potencial de energía solar que tienen los techos de las personas de las ciudades donde se hizo el mapeo, actualmente mapearon 7 ciudades en EU y una en Chile. El resultado que muestran es un análisis financiero y de eficiencia eléctrica sobre los paneles solares que se pueden instalar en los techos de cada persona. La manera en como lo realizaron es con la Metodología del SIG (sistema de información geográfica) y con un mapeo por medio de LIDAR en toda la ciudad, gracias a ello pudieron tener un análisis de sombra que detalla más sus resultados financieros, adjuntando algunos datos climáticos (Mapwell, MIT & Consultora MODE, 2013).
- El proyecto de investigación “Estimación del potencial de energía solar de Hong Kong’s usando SIG y tecnologías de la teledetección”, el proyecto llevado a cabo por el equipo del Departamento de Topografía y Teleinformática. La investigación utiliza, tecnologías de teledetección (RS), sistemas de información geográfica (SIG) y tratamiento de imágenes digitales, para la medición de cuantos paneles se podrían instalar en todos los techos de la ciudad, analizando el potencial de energía solar, de igual manera sus resultados muestran un estudio de sombra entre edificios, además de dar un estudio completo de nubosidad como

afecta a la producción anual de la simulación de paneles solares de la ciudad. El resultado final es la producción de electricidad de paneles fotovoltaicos de Hong Kong que podría dar en sus techos es de 2.66 TWh (Tera watts) (Sing Wong, Zhu, Liu, Lu & Peng, 2016).

- EL “Proyector Sunroof” de Google, es una plataforma abierta que abarca 43 millones de techos en EU, disponible en 50 estados, donde el usuario puede seleccionar su techo mostrando cuanta producción eléctrica por medio de paneles solares puede generar el techo y cuanto dióxido de carbono estaría ahorrando al planeta, y vincular al usuario con los pequeños instaladores que estén en la localidad. Este resultado logra con los datos de Google Earth, realizar modelos 3D que simulen el sombreado y el espacio que se requiere para su instalación. Los modelos 3D estiman la cantidad de espacio disponible para paneles solares, incluido el paso y el acimut de cada plano del techo, y en particular los contornos para separar los techos de otros objetos, se utilizó una red neuronal profunda. Y para lograr eficiencia incluye los datos climáticos típicos, recolectados por la misma compañía Google (Google Inc, 2015).
- El proyecto de investigación, “SIG para análisis socio-energético hacia un panorama renovable en Cancún”, el trabajo propone un SIG para el desarrollo de análisis socio-energético. Haciendo una descripción geográfica, demográfica, de comportamiento del consumo energético y el potencial de las diferentes energías renovables de la localidad. El SIG incluye como variables de entrada: cartografía catastral de la ciudad, modelos de elevación, cartografía del potencial energético renovable, variables demográficas, consumo y distribución de la red eléctrica, estos representados en formato vectorial y ráster, esto permitió generar variables derivadas sociales y energéticas, como resultado propuestas que sirvan de apoyo para tomadores de decisiones en los sectores público y privado. El resultado de toda la investigación muestra mapas de áreas factibles para la instalación de paneles fotovoltaicos en techos de toda la ciudad, turbinas eólicas y hornos de biomasa como generadores eléctricos dentro de la ciudad, y más mapas derivados contrastando con variables sociales (Gean C. et al, 2016).
- La investigación “Deep & Dense Convolutional Neural Network for Hyperspectral Image Classification” desarrollo una herramienta para la clasificación de imágenes hiperespectrales (HSI) con detección remota bajo las redes neuronales convolucionales (CNN), creando una arquitectura con menor margen de error al momento de la clasificación. Los datos que se utilizaron para el entrenamiento y su evaluación, fueron extraídos de una base de datos abierta, los datos fueron alrededor de cincuenta mil datos segmentados y clasificados. Este modelo que propusieron agrega conexiones internas entre bloques ocultos en las capas convolucionales para propagar hacia adelante todos los mapas de *features*, obtenidos en todas las capas que componen la red. Con ello son capaces de discriminar y mejorar las características contenidas en el HSI original. Creando así una herramienta poderosa para el análisis de imágenes hiperespectrales de grandes y complejos conjuntos de datos (Chang et al., 2019).

- El proyecto de investigación “Segmentación de imagen del suelo y análisis de textura: Enfocado por visión de computadora”, tiene por objetivo, el procesamiento automatizado de imágenes digitalizadas de secciones de suelo, para la estimación, de informaciones biológicas y los cambios ecosistemas terrestre, como también la extracción de características, el análisis de textura y segmentación en regiones homogéneas. La forma en como lo lograron fue segmentarlo por regiones morfológicas basado en el crecimiento de las mismas regiones y la velocidad del cambio del contraste. Luego se analizó la textura con filtros de frecuencia múltiple, operadores de seguimiento de energía (concentración de que tan homogéneo es la textura en una sección de la imagen) y las estimaciones de la modulación espacial. Los resultados experimentales muestran una eficiente de segmentación, en las diferentes secciones de suelo, a imágenes microscópicas como macroscópicas, lo enriquecedor de estos resultados es que también aplicaron para imágenes satelitales de ciudades donde se observa una segmentación de fácil interpretación (Sofou et al., 2005).

Todos los proyectos mencionados son los más relevantes encontrados en la literatura para la presente investigación. Cada uno presentan técnicas como metodologías que sirvieron como referencia, estructuras a emular y comparación de resultados en la investigación. Cabe señalar que cada uno tiene sus requerimientos para llegar a sus resultados obtenidos, es decir, el proyecto Mapwell, el SIG de Hong Kong y el Proyecto Roof de Google, requieren una obtención de datos con un costo alto en cuestión de recurso monetario (hardware especializados) y de tiempo, los tres proyectos mapearon las ciudades con diferentes tecnologías para luego poder procesar la información y representar el recurso solar, por ello sus métodos no son replicables en diferentes ciudades, aunque sus resultados son más precisos teniendo una relación directamente en que tan rigurosos fue el mapeo, siendo este una desventaja para la replicación en otras ciudades. De manera semejante ocurre con el proyecto de clasificación hiperespectral de imágenes con CNN, este proyecto obtiene capas de clasificación en su arquitectura de la red que segmentan con precisión los techos como otras partes de la ciudad (parque, estacionamiento, calles), estos algoritmos son muy enriquecedores porque la red neuronal profunda al extraer sus propias características, la mayoría son de textura y otras variables, esto es equivalente al método que utilizaron para segmentar regiones heterogéneas en la segmentación de imágenes de suelo, dando un recurso fuerte para la segmentación. El proyecto de investigación de segmentación de imágenes de suelo no requiere una obtención de datos costosa, ni una cantidad grande de datos, la clasificación de diferentes tipos de tierra como sus nutrientes lo realiza con las mismas imágenes y se nutre a sí mismo cuando le presentan más secciones de la imagen, segmentándolo con su propia información, esta investigación tiene una gran influencia para la sección de segmentación del algoritmo realizado en la presente tesis. En resumen, las investigaciones y los proyectos mencionados, dan pauta a la innovación y evoluciones para el desarrollo de un modelo integral, con bajo costo de obtención de datos y sin requerir una cantidad grande de ellos, con el factor que se pueda replicar en diferentes ciudades semejantes.

En efecto de todo lo anterior se ve una necesidad de crear un modelo para México, con una base de datos abierta y con ciudad que sea el caso de estudio tanto para la comparación y validación de sus resultados, en esta investigación sería Mérida por la accesibilidad de sus datos. En el contexto local, en México no existe un modelo para la transición energética que muestre el potencial de recursos renovables especialmente solar, en ciudades, siendo un área de oportunidad con urgencia para gestionar el cambio y optimizar los recursos energéticos.

Por otro lado, existe un incremento de sistemas interconectados a la red, estos microsistemas domésticos están creciendo en diferentes ciudades, pero centrándonos en nuestro caso de estudio que sería Mérida de igual manera hay un incremento relevante en microsistemas que no tienen una gestión adecuada para una planificación de la ciudad a largo plazo, en el informe del 2018 de CFE (comisión federal de electricidad), el porcentaje de crecimiento fue de crecimiento del 2018 fue de 30 % conforme al año pasado<sup>1</sup>, y se espera el mismo crecimiento o mayor en los siguientes años.

Esto es favorable en la disminución de dióxido de carbono, pero logísticamente y términos de planeación es una desventaja, dado que CFE realiza su producción bajo el historial de la demanda, generando una sobre-producción, dado que no contempla como fuente de apoyo los microgeneradores domésticos, por ello es necesario tener una herramienta de planeación energética que mida el potencial solar de los techos para que CFE prevea las inyecciones esporádicas de la red de la ciudad, y cuantifique su generación, creando así planes factibles según la prevención de ello, en corto y largo plazo.

Otro punto importante que argumenta lo relevante de la presente investigación, son los megaproyectos de paneles solares en México, no son soluciones sustentables, por varias razones, las cuales son, una deforestación total para su instalación, no genera sombras como un segundo techo sobre las casas (disminuyendo la temperatura sobre la misma casa y por ende reduciendo el consumo eléctrico), existe un mayor margen de pérdida de eficiencia eléctrica, dado que se requiere que sea lejos de la ciudad existe una pérdida potencial en traslado de la energía desde donde se genera hasta donde se solicita su demanda, no es un generador de empleos de calidad en el sector, además de disminuir micro-economías emergentes como las empresas de instalación y mantenimiento de sistemas domésticos. <sup>1</sup>Es por ello que se ve una fuerte oportunidad en las áreas no utilizadas que son los techos de una ciudad, tratando de hacer una equivalencia de generación de los mega-proyectos contra la suma de techos la ciudad.

Por último, la georreferenciación de datos relevantes en una ciudad, apoya tanto a los tomadores de decisiones y público en general a la visualización con varios enfoques tanto en lo energético, social y económico, dando pie para nuevas investigaciones o propuestas por cómo se analizan los datos.

---

<sup>1</sup> Subsecretaría de Electricidad, "Estadísticas del sector eléctrico e Indicadores de CFE y LFC", Sector Eléctrico Nacional, <http://egob2.energia.gob.mx/portal/electricidad.html>

## Objetivos

El objetivo general de la presente investigación es desarrollar un modelo que muestre el potencial del recurso solar, de una ciudad basándose en fotos satelitales de la misma, creando un algoritmo de clasificación que determine los techos factibles para la instalación PV.

### Objetivos específicos

- Desarrollar un diagrama de flujo modular y escalable para mejorar los resultados de la investigación por partes del algoritmo, optimizando las pruebas de otros métodos.
- Desarrollar un algoritmo de segmentación que descarte todo lo que no sea techo, de tal manera que pre-procese la imagen satelital.
- Desarrollar una base datos confiable que sirve para el entrenamiento de modelo de IA para su validación.
- Diseñar métricas de evaluación de diferentes arquitecturas de aprendizaje de máquinas para elegir el mejor algoritmo de clasificación.
- Hacer un análisis de los *features* para el modelo de aprendizaje de máquinas.
- Desarrollar un algoritmo integrado que una la segmentación y la clasificación, con un post-procesamiento para una mejor visualización de la imagen.
- Analizar diferentes secciones de la ciudad, que tengan peculiaridades, para el análisis del algoritmo propuesto.
- Establecer el modelo propuesto, en un caso de estudio, para su respectiva evaluación y aplicación de resultados.
- Establecer, una evaluación experimental para realizar la comparación de resultados, lo estimado contra lo real.
- Desarrollar una integración a SIG para el desarrollo de un software beta para la selección de casa según el usuario, mostrando de esta manera la cantidad de paneles que se puede instalar por casa.

## Metodologia

La metodología del presente trabajo, se encuentra distribuida en 3 partes principales, como se puede observar en el diagrama 1, el modelo final, es el algoritmo integrado de estas tres grandes partes, lleva una forma modular es decir que la mejora en cada parte afecta en un mejor desempeño en las partes siguientes, pero en su cálculo o en algún caso especial no afecta ni hace que se detenga en las partes siguientes.

La metodología se diseñó para aplicación y modificación de diferentes métodos dentro del comportamiento de los resultados en su clasificación, la primera parte, se creó el algoritmo de segmentación con métodos de tratamiento de imágenes digitales, probando y modificando diferentes métodos hasta que obtuve una imagen segmentada eficaz de esta manera se pre proceso la imagen, luego se integró a la arquitectura de aprendizaje de máquinas siendo menos el procesamiento necesario y se mejore en la clasificación.

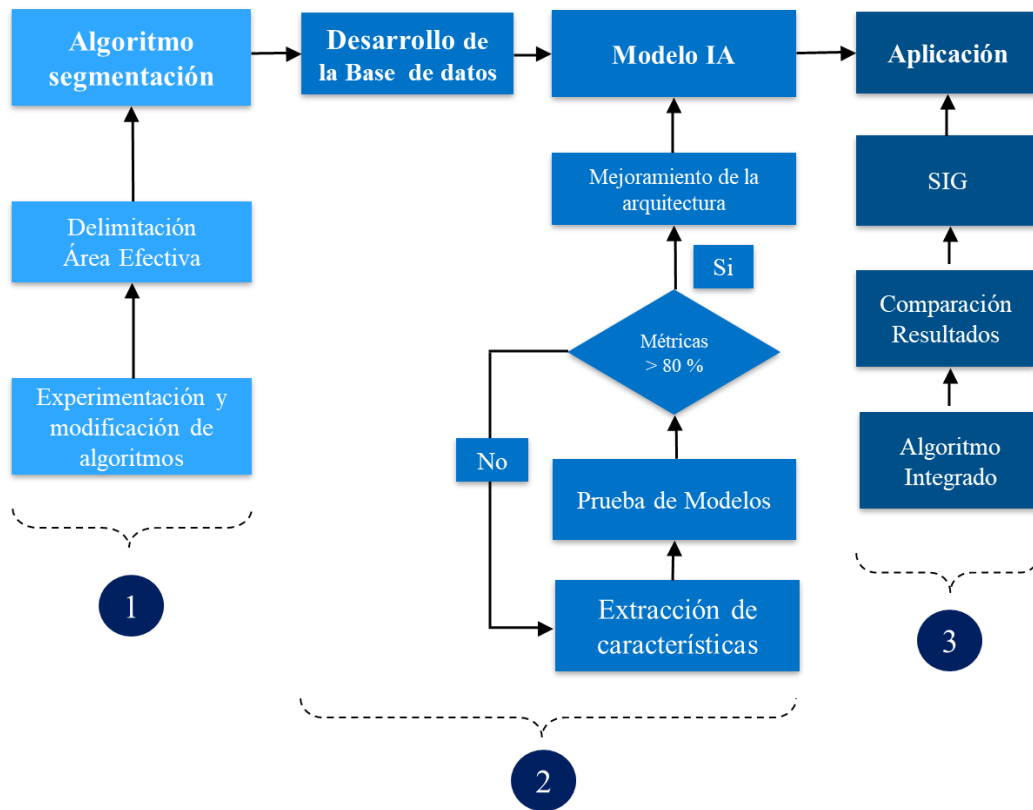


Figura 1 - Flujo metodológico

En la segunda parte, se construyó la base de datos de la ciudad de Mérida después de realizar una búsqueda exhaustiva de mapas de alta resolución o resoluciones que puedan extraer informaciones relevantes, se creó una base de datos de 200 muestras de techos y no techos, con validación de cada muestra, todo ello para el entrenamiento del algoritmo, se probó diferentes modelos de aprendizaje de máquina, cambiando con diferentes features hasta encontrar el que tenga mejor exactitud, luego se realizó un análisis de los features que son los relevantes para la clasificación.

En la tercera parte se integró los pasos anteriores con un post-procesamiento para una mejor visualización en un algoritmo integrado, luego se llevó a cabo un modelo experimental de resultados entre las estimaciones reales y lo que el algoritmo arroja, para poder hacer una cuantificación de los resultados, además de ver los casos especiales y su comportamiento de ellos, por último se realizó una integración con diferentes datos para mostrarlo con la Metodología SIG, con dichas bases se pudo realizar un software beta con el que el usuario pueda interactuar, el diagrama x, podemos observar como un panorama más global como se abordan estos pasos.

## Estructura de la Tesis

En el capítulo 2, se abordan los fundamentos de tratamiento de imágenes digitales, los algoritmos de procesamiento y post-procesamiento, luego se describen los métodos



utilizados para el desarrollo del algoritmo su teoría y la modificación que se realizó en ellos y como se fue mostrando los resultados obtenidos.

En el capítulo 3, se describen los fundamentos de aprendizaje de máquina, así también las métricas necesarias establecidas para la evaluación de los modelos utilizados, de igual manera la teoría y la obtención de los diferentes features. Además, como cada uno de ellos ayudo para la obtención del modelo presente, la descripción del mejor modelo con los features obtenidos como su representación matemática del modelo elegido.

En el capítulo 4, se presenta el algoritmo integrado, como se realizó la integración, los resultados de diferentes partes de la ciudad, un análisis de sensibilidad de cuáles son los peores casos para el algoritmo. Se realizó la comparación de ciertas casas muestras para la estimación del área factible de instalación que arroja el modelo contra la estimación de lo realmente se requiere para la instalación.

En el capítulo 5, se describe como se integró la metodología SIG para intersección de diferentes datos, el desarrollo de un software beta para estimación de producción de electricidad de casas seleccionadas por el usuario.

En el capítulo 6, se presentan las conclusiones de la investigación, los objetivos logrados y los trabajos futuros sobre ésta línea de investigación.

# Capítulo II

## Algoritmo de Segmentación

En el presente capítulo, se comenzará a introducir la primera parte del modelo integrado de clasificación de techos, que es el algoritmo de segmentación, el cual decanta todo lo que no es techo. Comenzando con el marco teórico, sobre el campo de tratamiento de imágenes digitales, desde cómo hacer la extracción de una imagen, hasta las metodologías estándar establecidas en este campo. Se explicará cómo se desarrolló el algoritmo de decantación, los métodos que se utilizaron y la aplicación en diferentes partes de la ciudad sobre este algoritmo.

### Fundamentos

La visión artificial y el tratamiento de imágenes digitales, ha tenido un fuerte progreso en investigaciones gracias a la evolución del equipo de cómputo tanto en la capacidad de procesamiento y en la memoria.

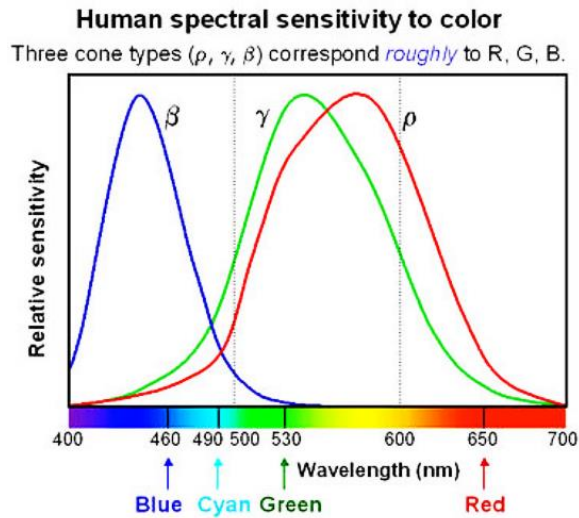
Es una disciplina compleja y multidisciplinaria que involucra otras ciencias, la visión artificial intenta imitar la capacidad de ver una escena y entenderla, al entendimiento de ello nos referimos, al procesamiento, interpretación o transformación de la imagen.

Por ello en las siguientes subsecciones, mencionaremos y explicaremos los fundamentos necesarios para poder entender los métodos que se utilizaron en el algoritmo de segmentación.

#### El Color

Para entender la abstracción de una imagen tenemos que comenzar con la representación del color. El color es un tributo que percibimos de los objetos cuando hay luz, los seres humanos distinguimos siete colores del ojo humano, y según la longitud de onda, distinguimos los tipos de luz, infrarroja (IR), visible, ultravioleta (UV) y blanca. Todos los cuerpos adsorben y reflejan colores, por lo tanto, lo que visualizamos es el reflejo de luz en los objetos (Gomez Allende, 2017). Como podemos observar en la siguiente gráfica, lo visible para el ojo humano siempre está en función de la exposición a luz y la longitud de onda, la ventaja es que el computador puede interpretar diferentes ondas.

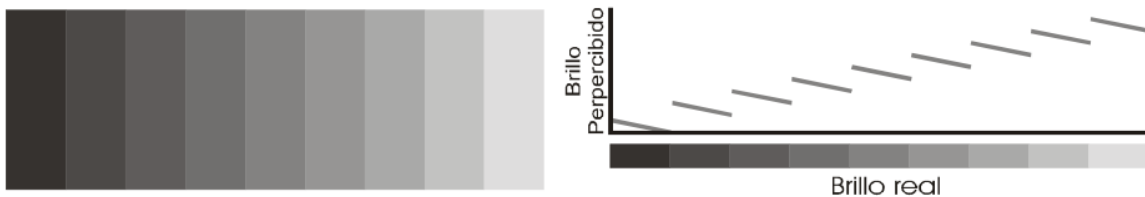
Las mezclas de los colores primarios producen todos los demás colores, los cuales son el rojo, verde y azul (RGB), sus valores van entre 0 y 255 (notación decimal) o entre 00 y FF (notación hexadecimal). También existen los colores pigmento (magenta, cyan y amarillo).



*Figura 2 - Color Sensible del Espectro Humano*

De igual manera, según los tipos de luz en función del color son, acromática, sin color intensidad o cantidad de luz, cromáticas, con color como radiancia (energía que sale de la fuente luminosa), luminancia (energía que percibe el observador, que procede de una fuente luminosa), brillo, la luminosidad de un color.

Un ejemplo de estas funciones son la inhibición lateral, es la perturbación en las fronteras del cambio de intensidad, esto lo podemos ver en la gráfica de la Figura 3, aquí podemos observar que la tonalidad de cada una de las franjas verticales de la figura de la izquierda es uniforme, sin embargo, al observarlas, parece que son más oscuras, por la derecha y más claras por la izquierda, pero para una interpretación más precisa, es que el brillo percibido para cada banda se refleja en el diagrama de la derecha.



*Figura 3 – Inhibición Lateral (Gómez Allende, 2017, pp.30- 35)*

Es importante el uso del color dado que es un descriptor poderoso, ello ayuda a una identificación de objetos y los extrae de la escena para luego, analizarlos. La imagen satelital de una ciudad esta compuestas por colores que con la interpretación al ojo humano clasifica ciertas partes la ciudad por intuición, con la misma lógica se base al algoritmo de segmentación.

Continuando con los fundamentos teóricos, para la estandarización de la interpretación de una imagen, existen modelos de color:

**Modelo RGB:** compuesto por componentes primarios rojo, verde y azul. Las imágenes que representan en este modelo tienen componentes, el número de bits que representa cada píxel en el espacio RGB, es llamado profundidad de píxel, una imagen de 8-bits tiene 24 bits de profundidad por las 3 matrices paralelas, cada píxel es una tripleta de valores de (R, G, B), la combinación de ellos nos daría el color del píxel. Como se puede ver en la siguiente figura, se puede observar en la imagen que cada cuadro es un píxel, cada uno de ellos según el modelo se componen con la tripleta de valores RGB en su notación decimal de 0 a 255, y también mostramos como el computador en código binario lo representaría.

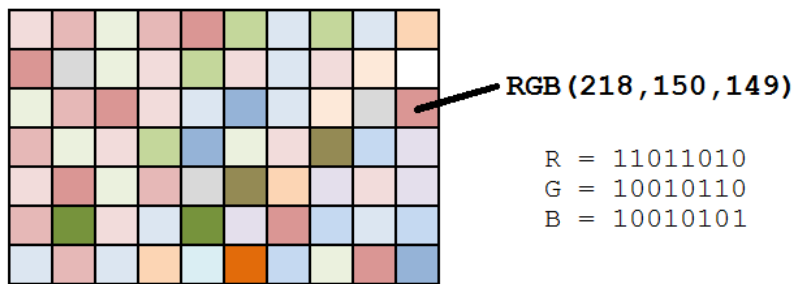


Figura 4 – Mapa de Bits (Gonzaga, 2015, pp. 20-25)

De igual manera la representación, de la imagen para el computador es la matriz de tres dimensiones, cada dimensión es la notación decimal en RGB, como podemos observar en la siguiente figura, con ello podemos sacar diferentes métodos que tratan la imagen para la obtención o la clasificación de los objetos.

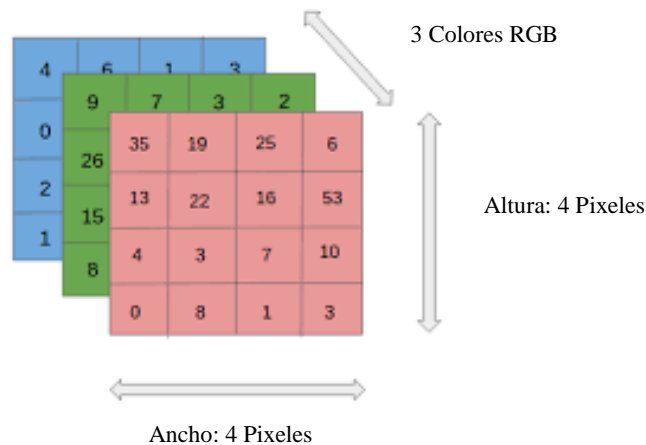


Figura 5 – Representación Matricial Modelo RGB

**Modelo CMYK:** el cian, magenta y amarillo, los colores primarios de los pigmentos. Para hacer una conversión interna de RGB a CMY se debe incluir un cuarto color K que representa el negro.

**Modelo HSI:** útiles para la interpretación humana, compuesto por tono (H), saturación (S) y brillo o intensidad (I).

Por otro lado, existen en las propiedades del color, las cuales hacen variar su aspecto según la apariencia final, son el tono, saturación y brillo.

Con todo lo anterior la transformación de una imagen de color a otra imagen de color, incrementa la percepción de los detalles de la superficie a resaltar, ha esto se le llama falso color. De igual manera existe otra técnica que utiliza los tonos de grises y realiza una transformación para generar en color, para una mejor interpretación del usuario, un ejemplo de ello lo podemos, en la siguiente figura, en esta figura podemos observar cómo, de imagen de escala de grises la transformamos a una imagen con un pseudocolor, de esta manera el usuario puede interpretar de mejor manera la imagen con los elementos a mostrar

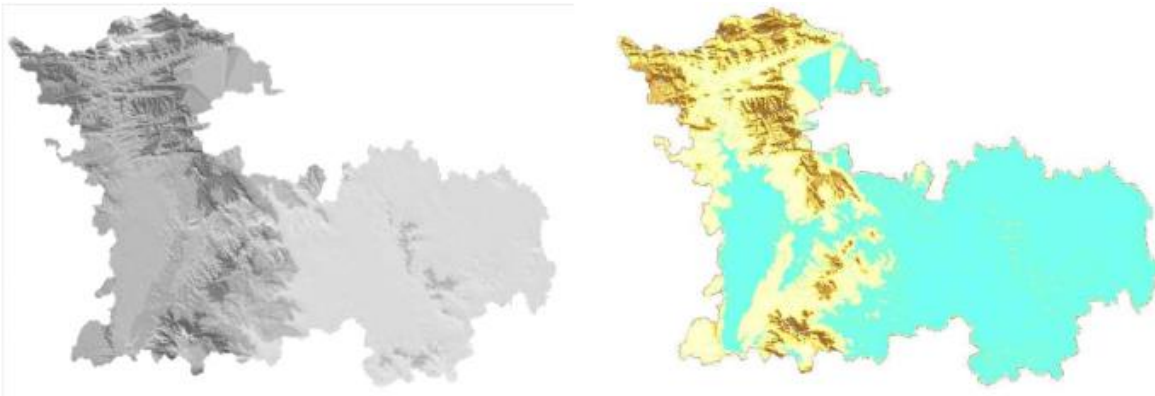


Figura 6 – Ejemplo de Transformación de Color

## La Imagen Digital

La imagen digital se define como la función bidimensional  $f(x, y)$  donde  $x$  y  $y$  son coordenadas en el plano y la amplitud  $f$  es la intensidad o el nivel de gris, cuando  $(x, y)$  y  $f$  son todos finitos (cantidades discretas) llamamos la función como imagen digital. Esto quiere decir que está compuesta por un número finito de elementos llamados *pixeles*, cada uno con un valor (relativo a una propiedad del color) y una posición en particular. Para detallar un poco más el pixel es la unidad mínima de información de una imagen, estas se componen por tres registros de color, mediante la combinación de rojo, verde y azul que adopta un color particular. Las imágenes bidimensionales son las proyecciones en perspectiva de escenas tridimensionales. Un ejemplo de ello es como se muestra de la ilustración 2 y 3 anteriormente.

Las imágenes digitales se pueden clasificar en dos tipos:

- imágenes vectoriales, conservan su nitidez de los bordes y no pierden detalles cuando se modifica el tamaño independiente de su resolución, la imagen no varía conforme al tamaño, es el tipo adecuado para el diseño de líneas, polígonos y figuras.
- Imágenes ráster o mapa de bits, número fijo de píxeles, dependen de la resolución, pueden perder detalle y *pixelearse* (notar más los píxeles) cuando se amplían, la principal ventaja es la posibilidad de recoger una gama tonal de información, el número de celdas de la imagen permanece variables, existen diferentes formatos para esta tipo de imagen, TIFF ( Tagged Image File Format), GIF (Graphic Interchange

Format), JPEG (Join Photographic Expert Group) y PNG (portable network graphics).

Para el presente modelo, se utilizará la imagen digital de mapa de bits o rastér, con formatos JPEG o PNG, dado que son archivos que tienen gama alta de calidad de imagen además de no se limita a 256 colores.

Existen diferentes tipos de imágenes digitales los cuales son, imágenes RGB, con tres canales para reproducción de colores en la pantalla, 8bits por canal, son de hasta 16,7 millones de colores. Las imágenes indexadas, reducen los colores de la imagen, a un máximo de 256 eliminando información del color. Imágenes en escala de grises se enfoca en los distintos tonos de grises esta imagen, se compone de 8 bits, cada pixel vario el brillo entre el 0 (negro) y 255 (blanco). Las imágenes binarias tienen una profundidad de color de 1 bit. Estos cuatro tipos de imagen se pueden en siguiente imagen:



Figura 7 - Tipos de Imágenes Digitales: a) RGB, b) Indexada, c) Escala de Grises, d) Binaria

Para el algoritmo como para el modelo, estaremos utilizando Imágenes bases e imágenes resultantes, las imágenes bases serán, imágenes RGB, imagen en escala de grises e imágenes binarias, cada una da un enfoque diferente para la segmentación de todo el modelo.

Estas imágenes tienen ciertas propiedades como son, la calidad de una imagen es determinada por el tamaño del pixel, que va por el rango el cual la captura la imagen. muestra la resolución espacial, un intervalo más corto produce una resolución espacial más alta. El brillo, entre más bits se utilicen más alta es la resolución del brillo. El ruido, todas las imágenes tienen cierta cantidad de ruido, se manifiesta como pixeles aislados que toman un nivel de gris diferentes al de sus vecinos y aparece como pequeñas variaciones aleatorias en el brillo y el color. Los tres factores nombrados anteriormente, hacen que variar la calidad de la imagen.

La resolución de una imagen es el número de pixeles que contiene una imagen expresada, el tamaño de la imagen es la cantidad de pixeles horizontales por la cantidad de pixeles verticales, como por la profundidad del brillo (bits).

#### Digitalización de imágenes

Es el paso de un mundo continuo (analógico) al mundo discreto (digital), al representar una imagen bidimensional, cada pixel ofrece información. En imágenes de blanco y negro es el brillo, pero en imágenes de color es la intensidad de componente del color, uno de los procesos fundamentales es:

El muestreo, es el proceso de obtener la imagen, en poner la imagen con  $m$  filas y  $n$  columnas, este determina el tamaño del pixel y el valor del brillo, y la calidad de cada imagen va depender del tamaño del pixel, entre más secciones se divida la imagen en diferentes pixeles, más alta va ser la resolución del pixel. En la siguiente imagen es la representación matricial de una imagen en tres dimensiones, siendo 1 la parte roja, 2 verde y 3 azul.

$$I_R(m,n,1) = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \quad I_G(m,n,2) = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \dots & g_{mn} \end{bmatrix} \quad I_B(m,n,3) = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

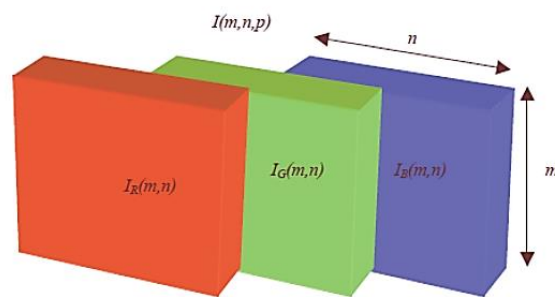


Figura 8 – Muestreo de una Imagen RGB

## Captura de imágenes por medio satélites

Existen diferentes formas para la captura de las imágenes, con varios hardware desde, cámaras fotográficas, cámara oscura. Pero existe la captura en la teledetección la cual es capturar una ciudad por medio de una imagen. Existen dos tecnologías que son confiables para la obtención de estas imágenes, una de ellas es la tecnología tipo LIDAR (*Light Detection and Ranging*), la cual consiste en escanear la ciudad por medio de un dron o un avión, a menos de 100 metros de altura de la misma ciudad, esta utiliza un sensor que emite un rayo para puntear y la respuesta que regresa es la profundidad de la superficie del terreno como de la elevación de los objetos contra el mismo terreno.

La segunda tecnología y es la que se utilizara en esta investigación, es la captura de la ciudad por medio de un satélite, que regresa una imagen tipo ráster. Una imagen satelital, se puede definir como la representación visual de la información capturada por un sensor montado en un satélite artificial. Este sensor recoge la información reflejada por la superficie de la Tierra enviado de regreso a esta y procesada. Existen diferentes tipos cuando es capturada, la imagen pancromática (color), monocromáticas (blanco y negro) y multiespectrales (información sobre diferentes bandas del espectro electromagnético) (Gonzaga, 2014).

Las plataformas (o satélite), los sensores de los satélites captan información para diferentes regiones del espectro y cada una de estas regiones se denomina canal o banda. La siguiente imagen muestra cómo funciona la captura de las imágenes por medio del satélite, con los dos tipos sensores que un satélite puede tener, sensores activos y sensores pasivos:

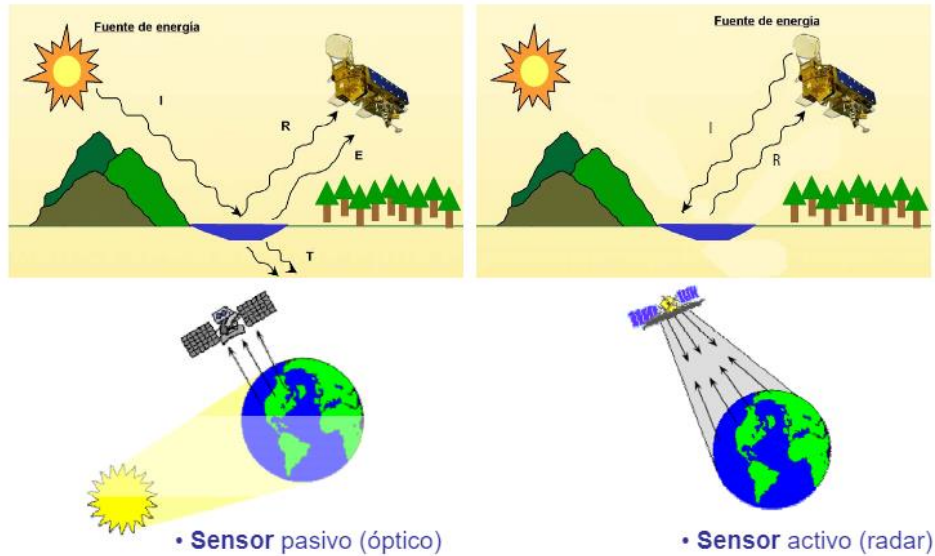


Figura 9 – Proceso de Captura de Imagen Digital (Ritendra et al, 2008, pp:25-30)

La salida de radiación (emitida o reflejada) de la superficie terrestre es un fenómeno continuo en 4 dimensiones (espacio, tiempo, longitud de onda y radiancia). Un sensor debe muestrear en este continuo discretizándolo en cada una de esas dimensiones. El modo en que esta discretización se lleva a cabo define los cuatro tipos de resolución con los que se trabaja en teledetección:

- Resolución espacial, este concepto designa al objeto más pequeño que se puede distinguir en la imagen. Está determinada por el tamaño del píxel, medido en metros sobre el terreno, esto depende de la altura del sensor con respecto a la tierra, el ángulo de visión, la velocidad de escaneado y las características ópticas del sensor.
- Resolución temporal (tiempo que tarda el satélite en tomar dos imágenes del mismo sitio)
- Resolución espectral, capacidad de discriminar entre longitudes de onda vecinas en el espectro, así como el número de bandas disponible.
- Resolución radiométrica, número de intervalos de intensidad que puede captar el sensor.

Una imagen de satélite en estado bruto es un conjunto de imágenes (una por banda) similares a las que se obtienen haciendo una fotografía digital. Los píxeles están organizados en un sistema de filas y columnas, pero sin referencia a coordenadas reales.



La resolución espacial es muy importante para la presente investigación dado que es la base de cuanta información podemos obtener y procesar, dado que existe imágenes satelitales de alta resolución como imágenes satelitales de baja resolución, en el caso de nuestro algoritmo como de la base datos de modelo de aprendizaje de máquina, se requiere imágenes satelitales de alta resolución no mayor a un metro. Un ejemplo de ello, es la imagen siguiente, como se puede observar, entre el tamaño de pixel este menor a medio metro, las características se pueden interpretar de una mejor manera, en la investigación si se tomara una imagen resolución de resolución mayor de un metro se perdería toda la información de la segmentación de los que es techo y de lo que no es, pero entre mayor resolución mejor se puede visualizar aun con el ojo humano para que se pueda segmentar.

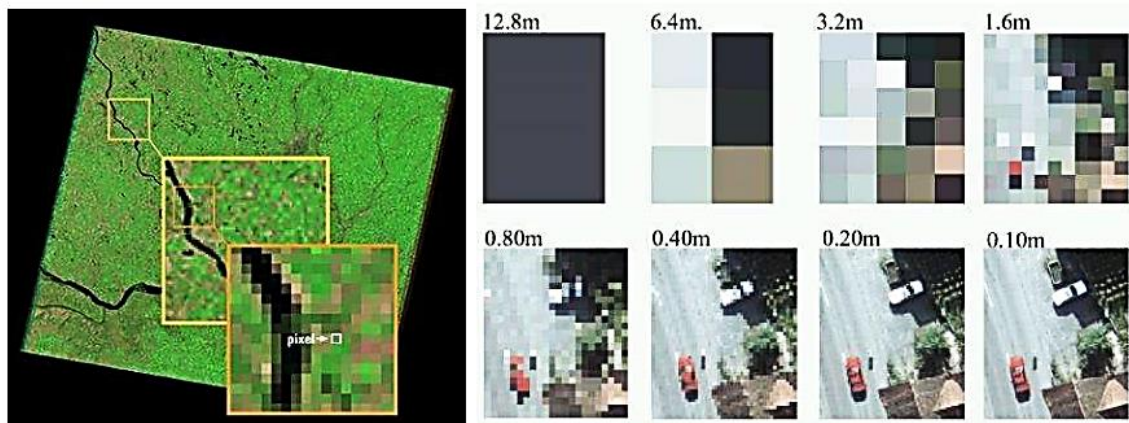


Figura 10 – Ejemplos de resolución espacial

Rafael C. Gonzalez, Richard E. Woods: *Digital Image Processing*. Addison-Wesley, Reading Mass 1992.

La obtención de imágenes de resoluciones espaciales de alta resolución, en especial para nuestro caso de estudio de la ciudad de Mérida, la mayoría de los empresa que otorgan estas imágenes actuales y de calidad, son de paga pero existe una empresa que por secciones y por el consumo de protocolo de HTTP (protocolo de transferencia de hipertexto) , se pueden usar las imágenes satelitales para la presente tesis como una base datos de calidad y actualizada, se detallara más adelante sobre la presente base de datos.

## Etapas procesamiento de imágenes

El procesamiento de imágenes digitales (PDI) o tratamiento de imágenes digitales, es procesar las imágenes del mundo real de manera digital por medio de un computador. Se

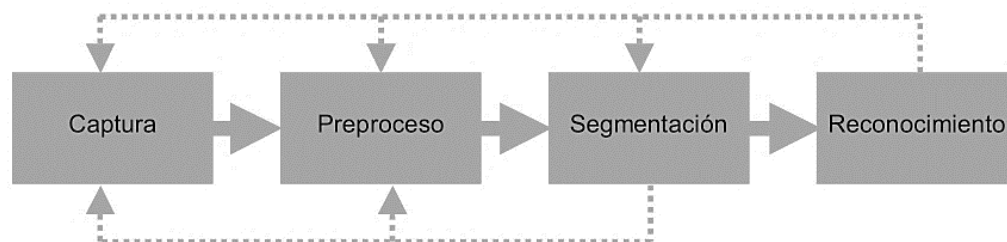


Figura 11 – Metodología del PDI (Velez Serrando et al., 2003, pp. 10–13)

caracteriza por la interpretación de información de imágenes digitales tomadas de un modo tridimensional. En la siguiente figura podemos notar un diagrama de bloques con etapas típicas en un sistema de visión artificial que es el campo del procesamiento de imágenes digitales.

En el anterior diagrama de bloques se puede describir con más detalle de la siguiente forma:

- La captura o adquisición: Obtener una imagen digital que tenga los requisitos necesarios para la extracción de la información.
- El preprocesamiento: reducir el ruido, realce del contraste, realce de ciertos detalles, o características de la imagen.
- La segmentación: Dividir una imagen en objeto de interes.
- La descripción: Obtener características convenientes para diferenciar los objetos de otros, formas, tamaño, área, estadísticas.
- El reconocimiento: Identificar los objetos.
- La interpretación: Asociar significado a un conjunto de objetos reconocidos, llegando al post análisis, y la obtención de datos.

Cada uno de estos procesos, son imágenes y sus salidas pueden ser desde imágenes o información de ella.

En general cada etapa del PDI tiene diferentes técnicas y depende directamente del problema, pero existe fundamentos teóricos que se deben de tocar para la realización de técnicas más avanzadas que integran de mejor manera estos fundamentos.

Se mencionarán estos fundamentos de manera general, los cuales son:

a) Relación entre pixeles

Un píxel  $p$  con coordenadas  $(x, y)$  tiene cuatro vecinos, dos horizontales y dos verticales, cuyas coordenadas son:  $(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y-1)$ ,  $(x, y+1)$ , esto se le llama vecindad 4 de  $p$  y se denota por  $N_4(p)$ , esto lo podemos ver en la ilustración 8 la matriz de la izquierda, existe una distancia de 1 pixel, y de igual manera existen los vecinos diagonales con coordenadas  $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y-1)$ ,  $(x-1, y+1)$  y se les denota por  $N_D(p)$ , como se ve la ilustración 9 de la figura de la derecha:

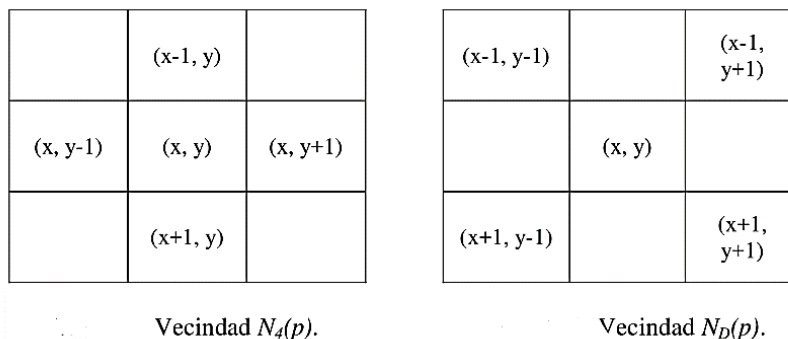


Figura 12 - Vecindad de una imagen Digital

b) Conectividad

Establecer los límites de objetos en regiones dentro de una imagen digital. Para determinar si dos pixeles están conectados ya sea por  $ND(p)$  o  $N4(p)$ , esto puede ser por la intensidad de gris, o si es binario o comparte el mismo valor en las bandas RGB todo ello en la misma vecindad.

c) Distancia o transformada de distancia

Proporcionar una medición de la separación existente entre dos puntos dentro de una imagen. Ponemos por ejemplo dos pixeles  $p$  y  $q$  con coordenadas  $(x, y)$  y  $(s, t)$  respectivamente, se pueden definir la función de distancia  $D$  si cumple:

- $D(p, q) \geq 0$
- $D(p, q) = 0$ , si  $p=q$
- $D(p, q) = D(q, p)$

d) Ruido

Todas las imágenes tienen ruido, depende la cantidad que cada una tiene, se representan como pixeles aislados que toman un nivel de gris diferente al de sus vecinos, creando muestras visiblemente heterogéneas, en la etapa de procesamiento del PDI, generalmente se eliminan el ruido de la imagen o se caracteriza, con filtros diferentes filtros. El ruido puede clasificarse en los siguientes tipos, gaussiano, impulsional, multiplicativo.

Otro fundamento y característica clave para la interpretación de la imagen es:

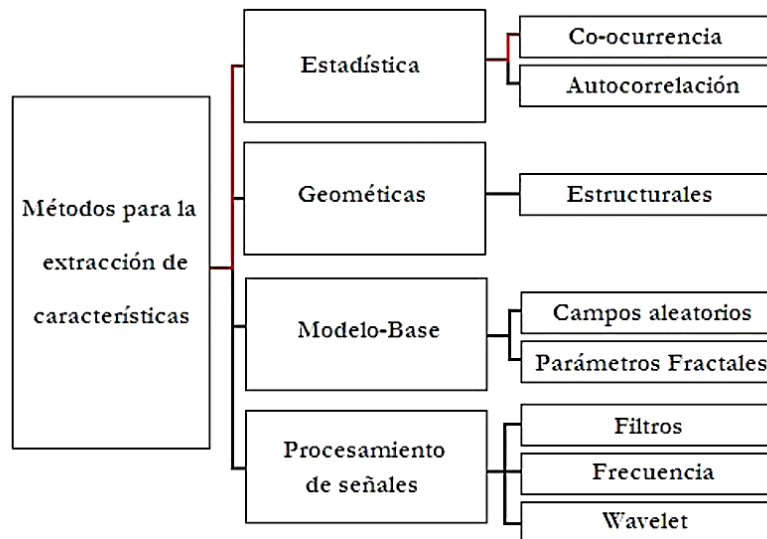


Figura 13 – Métodos para la extracción de características

## La Textura

Existen varias propiedades en la imagen digital, desde la medición de la geometría de objetos dentro de la imagen, el contraste y el brillo, pero la textura esta propiedad en especial para una imagen satelital nos permite diferencia varias secciones en la misma imagen.

En la siguiente imagen se demuestra una definición bastante figurativa, se define la textura como la repetición de un patrón espacial básico, cuya estructura puede ser periódica, o parcialmente periódica (aleatoria). Se puede caracterizar la textura de una superficie de una imagen, se toma en cuenta la distribución de valores de intensidad del color a nivel espacial, permitiendo evaluar diferentes propiedades.

En una textura formada por la repetición periódica de un patrón, la repetición es regular o estructurada, como por ejemplo dibujos en una tela y una pared de ladrillos, que se presentan en las imágenes 1 y 2 de la ilustración 4. En el caso de parcialmente periódica, el patrón se presenta de manera irregular o aleatoria, como por ejemplo tejidos de intestinos humanos, como las ilustraciones 3 y 4 (Le Seran et al, 2010).

Existen varias técnicas para análisis de textura, tampoco no existe una única teoría, y cada cual tiene un análisis único, estas se pueden clasificar en dos métodos estructurales y métodos estadísticos. Cada uno de los métodos lo utilizaremos en extracción de features para el modelo de ml (Machine Learning), y esas técnicas se detallarán de forma más precisas en el capítulo 3, pero mostraremos un cuadro que menciona los métodos más importantes que nos permiten para la extracción de características de la imagen, la línea roja muestra cual se utilizó en la presente investigación:

De todo lo anterior estos no son todos los fundamentos, del tratamiento de imágenes digitales, pero los mencionados anteriormente son claves para una comprensión más global de las siguientes técnicas con las que se trabajaron.

#### Análisis de técnicas de segmentación en proyecto de investigación

En esta sección nos enfocaremos, en el proyecto de investigación “Segmentación de la imagen del suelo y análisis de textura: Un enfoque por visión por computadora”, en especial en cómo se abordó la Metodología para esta investigación dado que, aunque no sea actual la forma en cómo se estuvo realizando la separación de cada uno de los puntos, fue innovadora por la integración de diferentes métodos utilizados.

El objetivo fue segmentar diferentes secciones del suelo, para segmentar utilizaron el método ‘morphological watershed transform’ con operadores no lineales morfológicos multi-escala, la manera en cómo se realizó fue: construir un gradiente de la escena de la imagen, obtener la región homogénea que se detecta, las líneas ‘watershed’ son asociadas, marcadas y construidas. Lo resultante de todo ello es como si la imagen tendría diferentes olas de texturas segmentadas, lo que resulta una segmentación por contraste de cada una de ellas (Sofou et al., 2005).

Esto importante, porque primero trataron de que cada región que desconocen se identifique sola por sus propios atributos, utilizando la textura para que la segmentación sea uniforme dando un umbral de cada región y el salto del texturas diferentes se identifique por otra región, luego de ahí utilizaron filtros no lineales para la interpretación biológica de cada región, este método en particular es poderoso dado que el objetivo de la investigación fue realizar un clasificar diferentes secciones de suelo, pero en sinergia lo que crearon a la par fue un algoritmo segmentado utilizando la propia textura la regiones interiores de la imagen

estableciendo un patrón en la regiones semejantes entre cada uno de ellas. En la siguiente imagen se muestra la aplicación de este algoritmo en una imagen satelital observado como el algoritmo segmentado por la textura da la misma imagen, un caso observable es el mar, dado que en la imagen se muestra como una textura casi uniforme de igual manera el algoritmo lo toma.

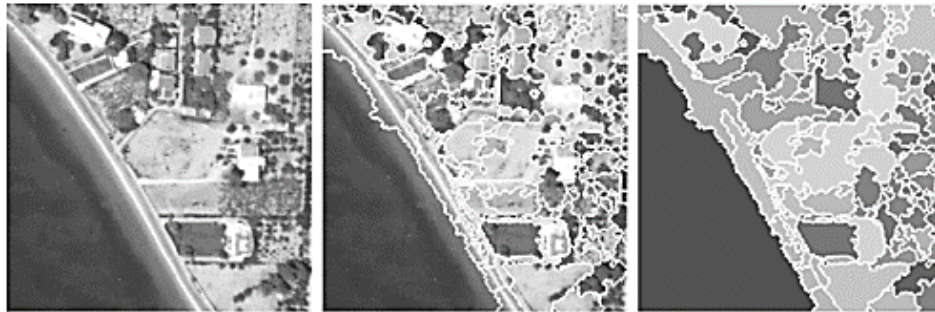


Figura 14 – Aplicación en una imagen Macro (Sofou et al., 2005)

Este proyecto, sirvió mucho por cómo se escogió la Metodología a seguir en la presente tesis por que no requiere más información solo que la misma imagen, y los atributos intrínsecos que vienen con la imagen satelital.

### **Metodología: Algoritmo de segmentación de la Imagen Digital**

En esta sección se hablará sobre el algoritmo de segmentación, la Metodología de este algoritmo se estuvo cambiando en varias formas y transformando parámetro de sus métodos, por cómo se aplicaba a la imagen resultante final con el algoritmo integrando dándonos este una importante retroalimentación tanto para el segmento del algoritmo de segmentación como para el modelo de *Machine Learning* que lo clasifica, por ello cuando se tuvo se comprobó que los resultados daban delimitación sin un margen de error grande se definió los métodos del algoritmo de segmentación.

En el siguiente diagrama muestra, flujo de la construcción del algoritmo de segmentación:

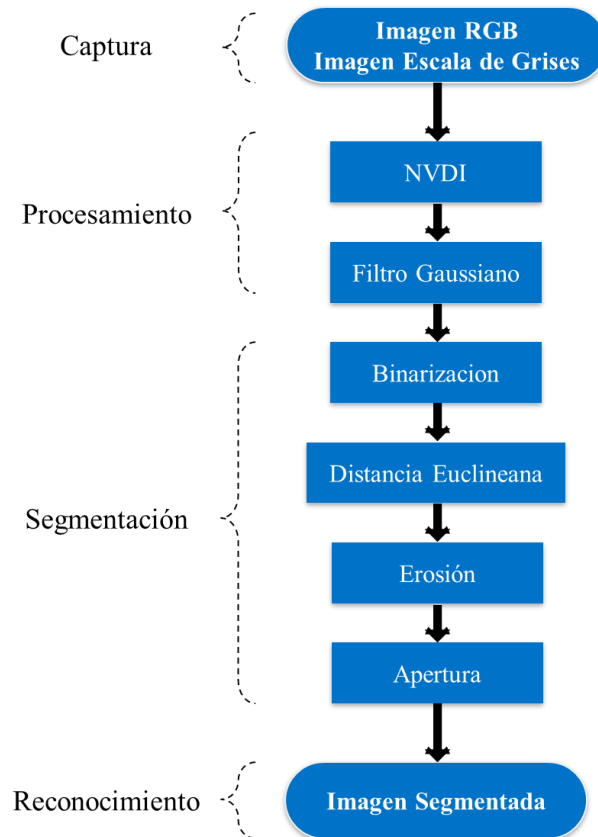


Figura 15 – Algoritmo de Segmentación por métodos

En el diagrama anterior y como lo explicado anterior en el subtema de etapas del PDI, se comienza con la captura u obtención de la imagen, que son las imágenes base, en los siguientes párrafos se detallara de donde se obtuvo la imagen satelital, luego procesamos la imagen con el primer método. El NVDI (índice de vegetación de diferencia normalizada) nos sirvió para la eliminación de toda la vegetación de la ciudad de Mérida. Luego se utiliza el filtro gaussiano para la eliminación de ruido de la imagen y el realce de la textura sobre las propias imágenes, obteniendo una salida de una imagen sin vegetación, con eliminación de píxeles inherentes y con realce de la textura. Con dicha imagen, se binarizaro bajo el método Otsu, es decir se pone un umbral para transformar a la imagen, en 0 y 1, esta binarización se hace con la conectividad del vecindario de la propia imagen sacando la textura, pero haciendo con un análisis estadístico, de esta manera nos muestra imagen binarizada que elimina vecindarios sin texturas predominantes, lo que arroja este método es una imagen con valores binarios. Con dicha imagen aplicamos la distancia Euclínea en uno de la imagen con la imagen, pero sacando el pixel de vecindario cercano del pixel del costado hasta que cambie su valor esto hace que los objetos dentro de la imagen se vuelan, objeto con vecindarios de texturas uniformes, se visualiza con más detalle en la sección del método del algoritmo.

Como últimos pasos se hacen la operación morfológica, primero la erosión, lo que realiza este método con la imagen que los objetos dentro de la imagen que tengan un valore diferentes lo rellena con el mismo valor del vecindario, haciéndolo reducir la imagen a estructuras geométricas, segundo se realizó la operación apertura, transforma la imagen de

tal manera que elimina los objetos pequeños menos de dos metros cuadrados y que no tengan la figura geométrica de un rectángulo, decantado de mejor manera de la imagen, la salida es una imagen binarizada, por último la realizamos una operación de multiplicación de imágenes y mullicamos la imagen RGB con la imagen binarizada.

En efecto de todos los pasos anteriores obtenemos una imagen segmentada, esta nos servirá para la clasificación con el modelo de *machine Learning*, la segmentación ayuda a la clasificación de los techos dado que además de quitar una carga de procesamiento, disminuye el rango de error del modelo al basarse en textura existen varios objetos que son en una ciudad que tienen semejanza con un techo factible de instalación.

### Obtención de imagen satelital

La obtención de mapas satelitales es por la empresa transnacional Esri (Environmental Systems Research Institute), actualmente desarrolla y comercializa software y proveer sistemas de información geográfica siendo una de las empresas líderes a nivel mundial en el campo del SIG. El producto que sirvió para fue la aplicación web, siendo más un servidor de consumo restirado, pero con datos abiertos de mapas ArcGis Online. Esta plataforma e agrupan varias aplicaciones para la captura, edición, análisis, tratamiento, diseño, publicación e impresión de información geográfica. Estas aplicaciones se engloban en familias temáticas como ArcGIS Server, para la publicación y gestión web, o ArcGIS Móvil para la captura y gestión de información en campo.

Las imágenes satelitales que obtenemos de esta base de datos abierta, son secciones la ciudad que se procesa para ver el comportamiento, cabe recalcar que el archivo del mapa de toda la ciudad no se descarga en ninguna máquina, toda la obtención del mapa para el tratamiento se hace bajo una petición GET de HTTP en el servidor de ArcGIS Online.

En la siguiente tabla mostráramos los detalles de la imagen satelital que estaremos trabajando:

Descripción de la imagen Satelital	
Servidor	Esri Community Maps Contributors
Satélite	GeoEye-1
Resolución espacial de la Imagen	0.5 m por pixel
Fecha de Captura	19 de marzo del 2019 a las 6:00pm
Tipo de Imagen	Pancromática

Tabla 1 - Características de Imágenes Digitales

La base de datos de imágenes satelitales se escogió por dos fundamentales razones, la primera que es la plataforma con la resolución espacial más alta siendo *open source* (es decir de código abierto para que cualquier usuario lo pueda consumir la información sin pagos). Segunda, es la más actualizada de entre todas las bases de datos consultadas. Estos dos puntos son primordiales pero el primer punto es necesario porque depende directamente la

resolución espacial de la imagen sea menor a sesenta centímetros por pixel, dado que si es mayor los detalles de cada imagen varían con ello la segmentación de la propia imagen se complica y la clasificación también, como resultado el algoritmo crea un margen de error grande y bastante perceptible al ojo humano, y el efecto es en sentido inverso de igual manera, es decir que entre mayor resolución se genera una mejor clasificación.

## Imágenes Base

Utilizaremos en todos los métodos del algoritmo, una imagen satelital base, siendo una sección de la ciudad, una imagen satelital del modelo RGB, esta porción de la ciudad se está mostrando con cada paso del algoritmo, la ilustración 10, es la imagen pancromática con el modelo RGB y es la imagen en escala de grises.

Cabe señalar que se está utilizando una sección de la ciudad al azar, en la última sección del presente capítulo estaremos aplicando más muestras con diferentes características para vislumbrar cómo se comporta el algoritmo de segmentación, sus ventajas y peculiaridades que se tiene del mismo.



Figura 16 – Muestra 1 Imagen Satelital RGB

Para la imagen en escala de grises, se utilizó la fórmula en la imagen de la ilustración 10:

$$Img_{gray} = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

La anterior fórmula de conversión es la misma que se utiliza para calcular la luminancia relativa en el espacio de color ITU-R BT.709 primario (el cual es mapa de colores de unión de primarios de RGB con los pigmentos) (Poynton y Charles, 2003). Este método de conversión es superior a los demás en imágenes satelitales por minimizar la pérdida de información de la compresión de la imagen.



La forma en cómo se transforma en escala de grises es que cada pixel se multiplica por las constantes que se presentaron en la anterior formula, pero en cada una de sus matrices de colores tanto rojo, verde y azul, de esta manera se transforma en una solo matriz, en escala de gris, cuando solo cada pixel solo tiene una profundidad y va desde 0 a 255. Como podemos ver a la siguiente imagen que será nuestra base en escala de grises:



Figura 17 - Imagen satelital en escala de grises

## NVDI

El índice de vegetación de vegetación normalizada o NVDI (*Normalized Difference Vegetation Index*), es un método de teledetección para estimar la cantidad, calidad y desarrollo de la vegetación con base a la medición, por medio de los sensores satelitales pueden ser activos o pasivos. Y funcion de esta manera se captura como la absorción de las plantas que emiten la radiación reflectante, generalmente se hace con imágenes satelitales multiespectrales, donde una de las capas es la información del IRCercano (infrarrojo cercano), emitiendo información sobre la absorción de las plantas en la región espectral de radiación fotosintética activa, la cual es usada como fuente de energía en el proceso de fotosíntesis. El infrarrojo cercano, es la región de la longitud de onda más corta del espectro infrarrojo situada entre el espectro visible y el infrarrojo medio.

Teóricamente la el NVDI de una imagen satelital se obtiene de la banda roja que es una de la matrices de la banda roja de la imagen, el IRCercano, una de las capas de la imagen multiespectral, y cada pixel se evalúa con la siguiente formula , luego de obtener el valor para cada uno, se establece un umbral dependiendo de la tabla de vegetación del NVDI, el cual es un hyperparametro, eliminando todo lo que no sea vegetación, la imagen satelital resultante es la capa de vegetación de la imagen (Gonzaga , 2014).

$$NVDI = \frac{(IRCercano - ROJO)}{(IRCercano + ROJO)}$$

Este método lo transformamos dado que nuestra imagen satélite no es de tipo multiespectral, si no pancromática, entonces utilizamos las bandas RGB de la imagen, como el siguiente diagrama de flujo refleja, y de esta manera se realizó la extracción de la vegetación:

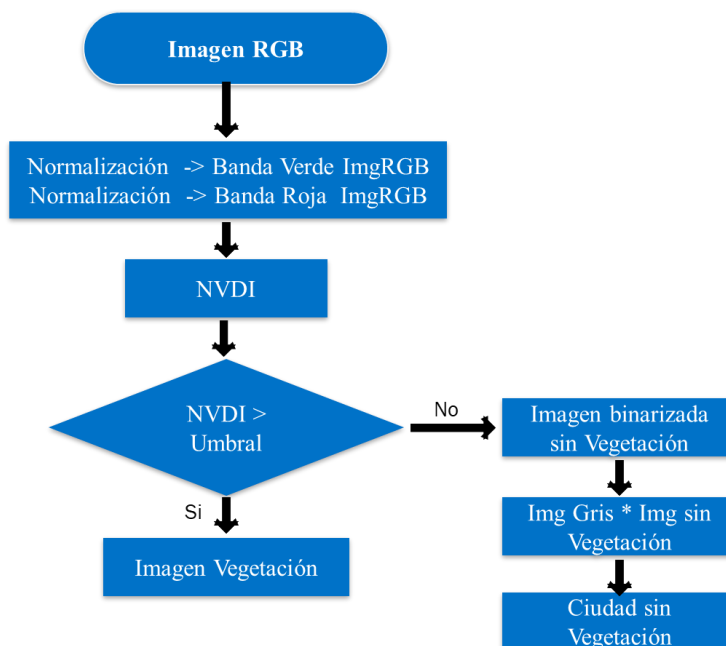


Figura 18 – Algoritmo modificado del NVDI

Primero de la imagen sobre mapa satelital, lo separamos en las tres bandas de colores que lo componen teniendo así un matriz de cada bando, para IRCercano utilizaremos la banda verde y para el rojo utilizamos la banda roja, pero a cada una de estas matrices primero se normalizara con la siguiente formula (Yeo y Johnson, 2000):

$$p' = \frac{p - \min(P)}{\max(P) - \min(P)}$$

La anterior formula es de normalización, se adecuo para que funcione en la de matriz donde  $p'$  es la proporción de pixel de la banda rojo o verde que de 0 a 1, y  $p$  es el pixel a tratar,  $P$  es la matriz que estamos tratando el  $\min$  es el pixel más pequeño de la imagen y el  $\max$  es el pixel más grande de la imagen, para cada banda va de 0 a 255, por el muestro de imagenes en valores hexadecimales, luego de ello con la fórmula del NVDI con cada pixel normalizado la banda verde normalizada haciéndose pasar el IRCercano y la banda roja normalizada, con ello tenemos una matriz con el NVDI, ahora tenemos que poner un umbral para la selección de la vegetación, se realizó bajo esto estándares, un ejemplo de esto es la siguiente tabla , que muestra dependiendo del tipo de vegetación que umbral debería de tener (Thorat et all, 2015):

Class	Minimum	Maximum
Water	-0.21	0.22
Settlement	0.02	0.18
Forest	0.33	0.55
Barren Land	0.07	0.36
Shrubs	0.14	0.31
Crop	0.36	0.65
Fallow Land	0.43	0.36

Tabla 2 – Índices sobre las clases de vegetación

Aun teniendo varias investigaciones que hagan referencia al umbral, todas recomiendan que es un parámetro que se debe de ajustar a cada caso con el que se está utilizando.

En nuestro caso el umbral escogido es de 0.57 se determinó por la tabla de referencia y por pruebas aleatorias a sección del mapa que mostraban mayor vegetación y la visualización era la retroalimentación de cómo se estuvo afinando el hyperparametro. Lo resultante de la selección del mapa, es el umbral, donde los pixeles que eran menores a este se establecían negros y los pixeles que eran mayores a este se volvían blancos, el resultado lo podemos visualizar en la siguiente imagen:

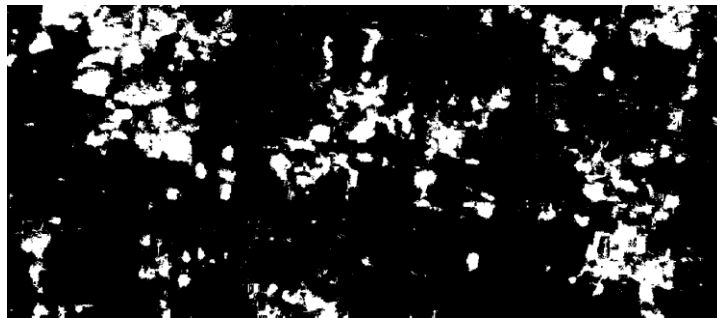


Figura 19 - Imagen Satelital solo de Vegetación

Por último, la imagen opuesta se sustrajo los valores de la imagen en escala de grises esto porque la imagen en escala de grises, se tenía una matriz con ninguna profundidad de igual manera la imagen binarizada, el resultante es la imagen sin vegetación:



Figura 20 - Imagen Satelital con NVDI aplicado

La anterior imagen de la parte derecha es la imagen sin vegetación, en diferentes secciones de la ciudad como en varias partes de la ciudad, la clasificación puede variar en muchos puntos de vegetación o no, pero esta etapa del NVDI no elimina la sombra de los arboles por ello alguna vez parece que no eliminara toda la vegetación, pero en realidad está eliminando toda la vegetación que la imagen pudo captar.

## Filtro Gaussiano

Este filtro se escogió como preprocesamiento de la imagen debido, a que estuvo probando con otros filtros siendo este el que daba más resultados en el algoritmo integrado, además que en la literatura recomiendan para imágenes satelitales este y el filtro de frecuencia.

Es filtro gaussiano se encuentra en el dominio del espacio, tiene como objetivo, modificar los pixeles cuyo nivel de intensidad es muy diferente al de sus vecinos, a la par de detectar aquellos pixeles donde se produce un cambio brusco en función de intensidad. Uno de los tipos de ruidos que se establecen en la mayoría es el de las imágenes satelitales son los ruidos gaussianos, esto se debe a la falta de iluminación o temperaturas, la intensidad del color se ve afectada, la siguiente grafica muestra como generalmente aparece el ruido, en las imágenes, sobre la parte superior de campana está el cambio de intensidad del pixel:

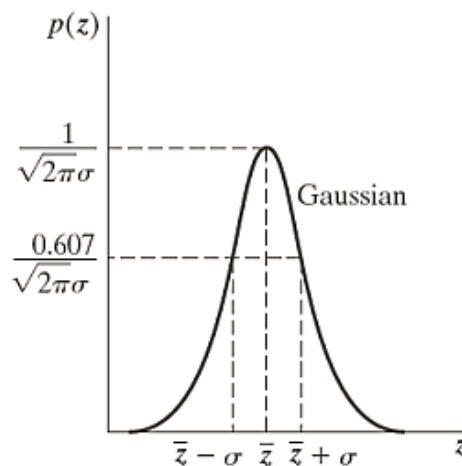


Figura 21 - Ruido Gaussiano (Thora et al, 2015, pp. 43)

El filtro funciona poniendo una máscara a cada pixel este se define por el tipo del filtro que va recorriendo la imagen pixel por pixel, la siguiente ilustración muestra de mejor manera como va recorriendo la máscara del filtro a nuestra imagen:

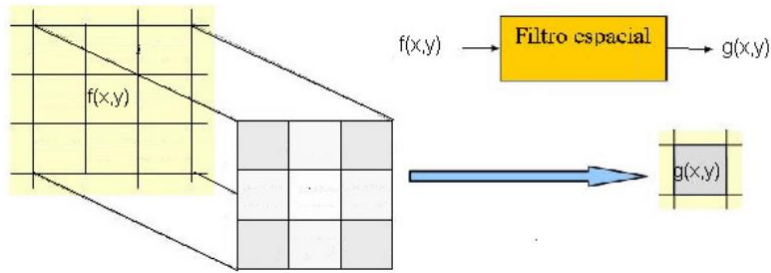


Figura 22 – Mascara de Filtro Espacial

La función matemática de la máscara del filtro es dada por, donde  $x$  y  $y$  son los pixeles de la vecindad que se genera en cada pixel:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Y es aplicado a cada uno de los pixeles, tiene dos hiperparámetros, que se establecen conforme al problema, el cual es determinar la sigma en la cual es de 0.5 y el segundo es determinar el tamaño de la máscara  $g(x,y)$  la cual sería una máscara de 3 columnas por 3 filas, y la máscara que obtenemos con estas especificación es, esto parámetros se tomaron gracias al algoritmo integrado por la retroalimentación general que se realizaba:

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

Tabla 3 - Mascara del Filtro Gaussiano

Aplicándolo a nuestra imagen de la sección de la ciudad sin vegetación, el resultado que se muestra sería:



Figura 23 - Imagen con filtro gaussiano aplicado

Ha simple vista, no se logra visualizar, la aplicación de la imagen. Por ello utilizamos el histograma como herramienta estadística, en la siguiente imagen, vemos el histograma de la

imagen sin el filtro, el eje de las y es el número de pixeles de la imagen y el de la x es la intensidad de grises de imagen como vemos esta agrupada entre valore de 30 y 220:

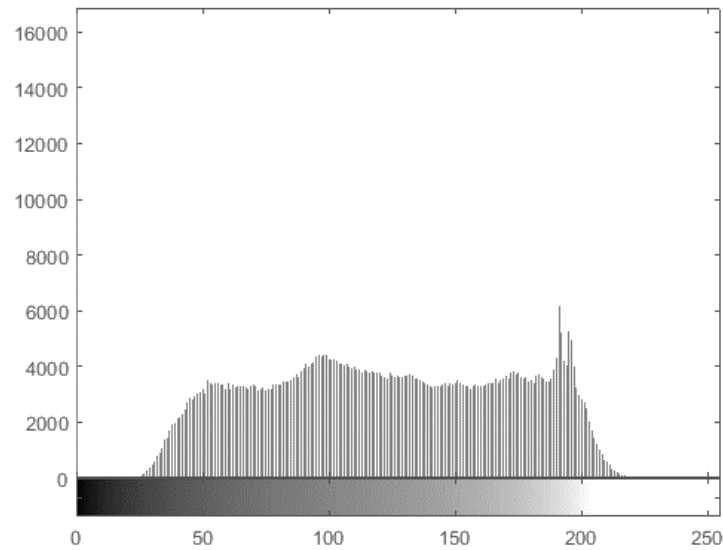


Figura 24 – Histograma de Imagen con ruido

Ahora veremos cómo aplicando el filtro se resaltan ciertas partes de la imagen satelital que, como lo valores de 0 que representan el negro. Este cambio se debe a que realza la frontera entre intensidades de vecindario, se realizara una detención de fronteras y un realce a sus texturas:

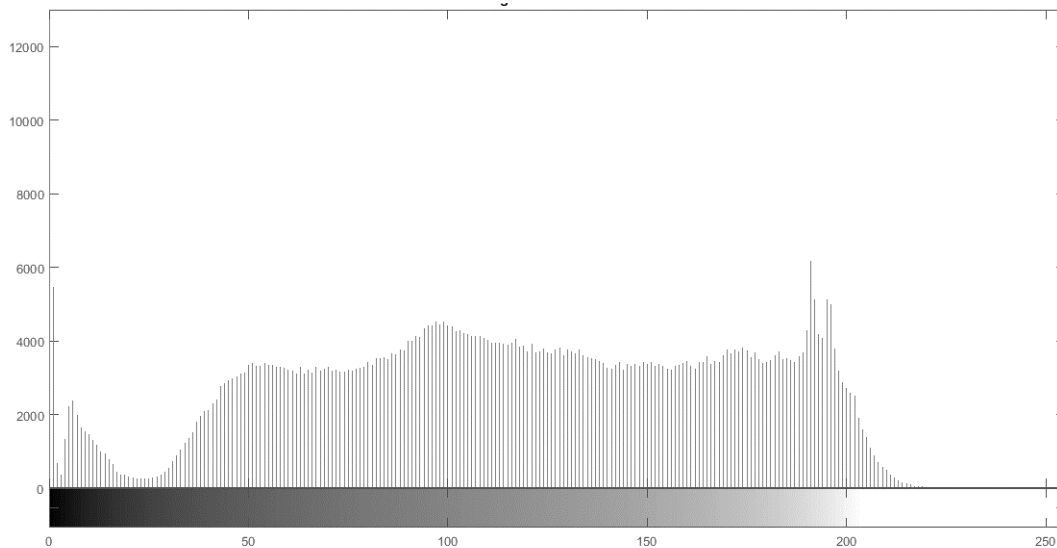


Figura 25 – Histograma de con el filtro gaussiano aplicado

De esta manera, este filtro nos ayuda a realzar las fronteras de una mejor manera, con ello podremos hacer una mejor segmentación de la imagen.

## Método Otsu

Luego tener la imagen con la eliminación del ruido, pasamos a la etapa de segmentación, en la mayoría de los proyectos de PDI, uno de los métodos más utilizados en la segmentación es dividir una imagen digital en regiones homogéneas con respecto a uno o más características, la propiedad que se busca es la discontinuidad, los cambios bruscos de nivel de gris, como puntos, líneas y bordes de la imagen.

El método que utiliza es establecer un umbral, para denotar los objetos de una imagen conforme a su fondo, con lo cual esto permite convertir una imagen de niveles de grises, de tal forma que los objetos de interés se etiqueten con valor distinto de los píxeles en el fondo. Existen varios métodos y también uno puede poner un umbral empírico hacer experimentación, llegar con prueba y error a un resultado, pero la mayoría de ellos son inexactos, por estas razones y también por cómo encuentra el umbral, se escogió el método Otsu.

El método Otsu, su ventaja primordial radica en que es automático, es decir, no necesita supervisión humana ni información previa de la imagen antes de su procesamiento, esto es clave para el algoritmo de segmentación porque no es una solución específica para una sección de ciudad, por que toma sus propiedades de a la misma imagen dada, lo que lleva a un método para diferentes casos adaptable.

En concreto el método Otsu, utiliza la varianza que es medida de la dispersión de niveles de grises, con el objetivo de calcular el valor del umbral de forma que la dispersión dentro de cada clase sea lo más pequeño posible, pero al mismo tiempo la dispersión sea la más alta posible entre clases diferentes. Por lo tanto, elige el umbral al reducir el mínimo la variación entre clases diferentes (Otsu, 1979). Se explicará de manera global como se algoritmo del método Otsu:

El método Otsu es una forma una búsqueda exhaustiva de buscar un umbral que minimiza la varianza intra-clase, se define como la suma ponderada de la varianza de dos clases en:

$$\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t)$$

Los  $w$  son las probabilidades de las dos clases separadas por un  $t$  (umbral) y  $\sigma$  son las varianzas de las clases:

La probabilidad clase se calcula a partir de los contenedores del histograma, donde  $L$  son niveles de gris:  $w_{0,1}(t)L$

$$w_0(t) = \sum_{i=0}^{t-1} p(i)$$

$$w_1(t) = \sum_{i=t}^{L-1} p(i)$$

Entonces Otsu muestra que minimiza la varianza intra-clase es la misma que la maximización de la varianza entre clases

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2$$

$$\sigma_b^2(t) = w_0(t)w_1(t)[\mu_0(t) - \mu_1(t)]^2$$

Se expresa en términos de probabilidades de clase y medios de clase:  $w \mu$ . Mientras la media de clase es:  $\mu_{0,1,T}(t)$  y la podemos expresar como:

$$\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{w_0(t)}$$

$$\mu_1(t) = \frac{\sum_{i=0}^{L-1} ip(i)}{w_1(t)}$$

$$\mu_T = \sum_{i=0}^{L-1} ip(i)$$

De las anteriores relaciones se pueden verificar fácilmente:

$$w_0\mu_0 + w_1\mu_1 = \mu_T$$

$$w_0 + w_1 = 1$$

Las probabilidades como las clases se calculan de forma iterativa (Rachel et al, 1992). Una forma en cómo se da esto, es el siguiente histograma que muestra cómo donde se pone el umbral después de hacer la interacción entre clases y clases internamente:

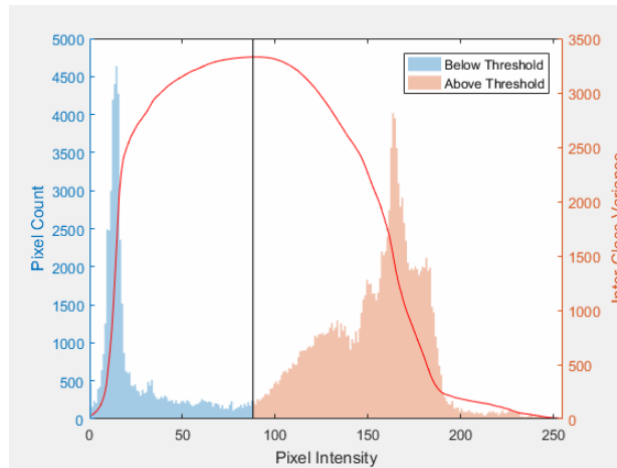


Figura 26 – Histograma Método Otsu (Sezgin, 2004, pp. 10)

En el diagrama 8 podemos ver que el eje horizontal de la izquierda es la cantidad de los píxeles y el eje horizontal de la derecha, en la varianza dentro de las clases que es la línea roja, en el eje vertical es la intensidad del píxel, luego de sacar el método Otsu con las ecuaciones anteriormente nombradas, sacamos el umbral que es la línea negra horizontal, este una gráfica solo de muestra para reflejar la separación.



Pseudocódigo del algoritmo:

1. Establecer histograma y las probabilidades de cada nivel de intensidad
2. Configuración inicial de  $w_i(0)\mu_i(0)$
3. Iteración, Paso a través de todos los umbrales posibles máxima intensidad  $t = 1 \dots$ 
  - a. Actualización del valor  $w_i\mu_i$
  - b. Calcular  $\sigma_b^2(t)$
4. Umbral deseado corresponde al máximo  $\sigma_b^2(t)$

Luego de ejecutar el algoritmo interior en la imagen el umbral que se tiene para la sección de ejemplo que estamos haciendo es: 0.3961

Con ello se aplica a la imagen y el resultado es la siguiente ilustración, que es la aplicación en la imagen en escala de grises:



Figura 27 - Imagen Escala de grises del Método Otsu

## Distancia Euclénea

Se utiliza la transformada distancia con el fin de calcular las distancias representativas que todos los píxeles tienen con respecto a un objeto en este caso los techos dentro de la imagen. Se probó con diferentes métodos, pero en los resultados no variaba significativamente, por lo tanto, se tomó el método que tenga el menor tiempo de cómputo.

Las transformada de distancia Euclénea se usa para definir de una mejor manera los contornos de los objetos dentro de la imagen, en este caso los techos, contra otros objetos esto lo realiza de la siguiente manera, dada una imagen  $(x, y)$  en una posición  $P(x, y)$ , cada elemento de la matriz tiene un valor que es la distancia del píxel representado a un contorno dentro de la imagen (Maurer et al, 2003). La fórmula es sacar la distancia entre dos puntos, pero del vecindario:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Se realiza de la siguiente manera

- Se hace un barrido de izquierda a derecha y de arriba hacia abajo en la matriz hasta encontrar el primer elemento con valor 0.
- A partir del primer elemento con valor 0, se sigue la misma dirección del barrido aplicando a los siguientes dos pasos a cada elemento siguiente con cada uno de los elementos vecinos que se encuentran arriba o a la izquierda, siempre que existan.
- Cada elemento vecino de los que se indican en el operador se le sumara el valor actual de la distancia que tiene el elemento como referencia, es 1 si horizontal o vertical o 1.41 si es diagonal
- Cuando se tiene todo el elemento se calcula el menor y se le asigna al elemento de referencia.
- Se realiza un barrido de derecha a izquierda y de abajo hacia arriba en la matriz ahora empezando por el último elemento de la matriz.
- A partir de este elemento se sigue la misma dirección del barrido, pero ahora de los se encuentren abajo siempre y cuando exista, como se ve en la ilustración x.

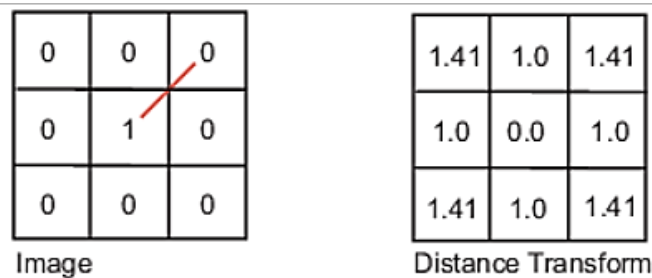


Figura 28 - Ejemplo de Transformada de Distancia Euclidean (Paglieroni, 1992, pp.10)

La matriz resultante será la transformada de la distancia del contorno de la imagen. El resultado de ello será la ilustración 15, que es la imagen resultante de los pasos anteriores más la aplicación de la distancia Euclidean:



Figura 29 – Imagen Satelital con la Distancia Euclidean Establecida

Como se puede visualizar en la imagen los contornos parecidos en las dos imágenes, sus transformadas serán similares, es decir, en los pixeles equivalentes habrá una diferencia pequeña en sus valores.

## Erosión

La erosión es una operación morfológica, que examina la estructura geométrica de una imagen usando como sonda un patrón de ajuntes, es decir su objetivo es que en una imagen binarizada los pixeles se vayan cambiando de tal manera que tomen el patrón del objeto a requerir (Gonzalez et al, 2009).

Se eligió este método dado que en tratamos de transformar los objetos que son posibles techos en objetos con una geometría más rectangular por la arquitectura que se tiene en diferentes casas de una ciudad.

La erosión funciona como la teoría de conjuntos de datos, pero en vez de ellos lo trabajamos con un patrón a seguir en toda la imagen. Su teoría nos dice que dada una imagen A, y un elemento estructura B, (ambas imágenes binarias), su erosión es el conjunto de todos los elementos x para los cuales B se trasladó por x conteniendo en A:

$$A \ominus B = \{ x \mid B_x \subseteq A \}$$

Tengamos en cuenta que, para la condición  $B_x \subseteq A$ , solo consideramos los pixeles negros A y B. La Erosión es la operación morfológica dual de la dilatación. La erosión se concibe usualmente como una reducción de la imagen original. Un ejemplo de este caso es la siguiente imagen donde tenemos el objeto a buscar y se recorre en la imagen disminuyendo para buscar el patrón:

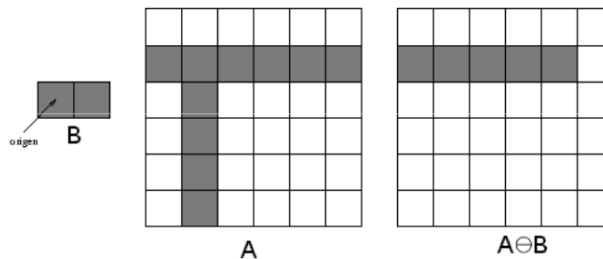


Figura 30 – Ejemplo de Operación Morfológica Erosión (Abad, 2016, pp.21)

Para nuestra implementación el objeto como patrón a recorrer un cuadrado de pixeles de dos por dos en una forma matricial sería  $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ , dado que queremos transformar la imagen a imágenes más rectangulares, por la forma en como son los techos, en la siguiente ilustración es el resultado de la aplican más los métodos anteriores realizados en este capítulo:



Figura 31 – Imagen Satelital con Erosión Aplicada

En diferentes partes, especialmente en las esquinas y también, en los coches este método elimina los que no sea un rectángulo con tamaño significativo buscando la forma definida, reduciendo la imagen a los objetos que estamos por buscar, los techos.

## Apertura

La apertura es una operación morfológica, que contiene el elemento estructural, es un método que generalmente suaviza los contornos de una imagen y elimina pequeños salientes, eliminando franjas o zonas de un objeto que sean más estrechas al elemento estructural (Gang Li, 2010).

Y se define como, imagen  $A$  por un elemento estructural  $K$ :

$$A \circ K = (A \ominus K) \oplus K$$

La apertura es  $A$  por  $K$  la erosión de estos dos como el método anterior seguido por la dilatación, donde se expande la imagen. Si en el caso que  $A$  no cambie con la apertura con  $K$ , se dirá que  $A$  es abierto respecto a  $K$ . Un ejemplo visual es el siguiente donde  $B$  es el objeto estructural y  $A$  es la imagen, se ve cómo cambia el contorno, rompe uniones estrechas entre partes de conjuntos y elimina salientes estrechos:

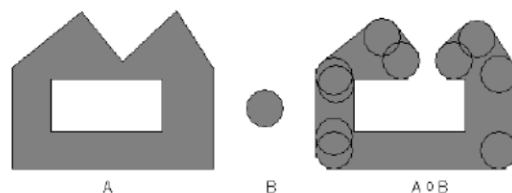


Figura 32 – Ejemplo del Método Apertura (Abad, 2016)

En nuestro caso el elemento estructural, es el cuadro de 2 píxeles por 2 píxeles, con ello después de hacer la erosión por sí sola, la mayoría de los objetos dispersos hace que realce más los techos con la forma rectangular que se tiene, y los puntos dispersos los elimine, la siguiente imagen es la imagen con método ya implementado y también la imagen con el algoritmo ya aplicado:



*Figura 33 – Imagen segmentada*

Como podemos ver se, puede ver en diferentes partes de la sección de la ciudad que se ha eliminado en la mayoría los puntos dispersos en toda la imagen, dejando los techos han estado teniendo una textura más completa, con ello comprando en esta sección de imagen que algoritmo de segmentación funciona.

## **Aplicación del Algoritmo Segmentado**

En las anteriores secciones, se pudo ver como el algoritmo de segmentación de techos, está compuesto, además de como con la misma información de la imagen va eliminando y moldeando todo lo que no sea techo, con este nuevo mapa resultante de la sección de la ciudad, se procesará luego con el modelo de aprendizaje de máquina, se podrá ver la clasificación de los cuales son los techos factibles y cuales no.

En esta subsección veremos diferentes secciones de la ciudad, para analizar cómo se comporta el algoritmo de selección, estas secciones son diferentes entre, pero es para demostrar como el algoritmo se comparte en diferentes casos, y cual es caso especial del algoritmo. No se tomó la ciudad completa dado que el servicio de mapas de ArcGIS solo da secciones de mapa y algunos mapas de la ciudad de Mérida que tienen una resolución espacial de 0.8 m por píxel o 1 m por píxel, aunque no tiene la misma información el problema recae, en el tiempo de cómputo y como se visualiza dado que no se puede apreciar a detalle cada casa, sin poder analizar cada caso especial, por ello expondremos los casos no convencionales para en la aplicación del algoritmo:

### **Caso 1**

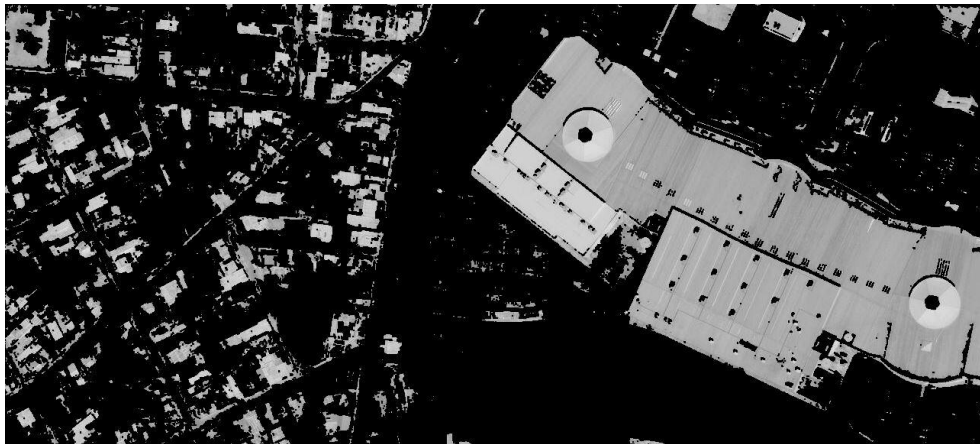
En esta sección colocamos, la parte de la ciudad pertenece a un edificio que es un salón de conferencia y alrededor se encuentran estacionamientos calles y arbolada. Una característica del algoritmo es que decanta todo lo que es vegetación de manera eficaz, pero en cuestiones

de asfalto y calles, de igual manera diferencia entre estos dos, como se puede ver en la ilustración de la imagen segmentada, es decir se puede ver como se diferencia entre los matices de concreto.

Esta muestra se escogió dado que el techo del edificio de conferencia tiene cierta semejanza en los niveles de grises de las diferentes partes de la imagen, además denotar que el algoritmo de segmentación encuentra una textura uniforme, en un área grande y no toca esa parte, por lo cual transformada de distancia y la binarización que se ocupó de este punto.



*Figura 34 – Imagen Satelital Caso 1*



*Figura 35 - Imagen Resultante Caso 1 Segmentada*

## Muestra 2

La siguiente ilustración muestran, es una sección de la ciudad donde son los límites de la ciudad, en este caso podemos visualizar el anillo periférico que rodea la ciudad, es una de las carreteras más anchas de las ciudades, como se puede ver en la ilustración. El algoritmo segmenta toda la textura uniforme rectangular, pero deja algunas secciones de carretera que es asfalto, un material parecido a la construcción de los techos, pero en general las carreteras el algoritmo no los toma como techos.

Otra característica que también pasa, es que el algoritmo de segmentación para comunidades hace notar de una mejor manera dado que la textura del techo se contrasta mejor, como se puede ver en la imagen resultante.



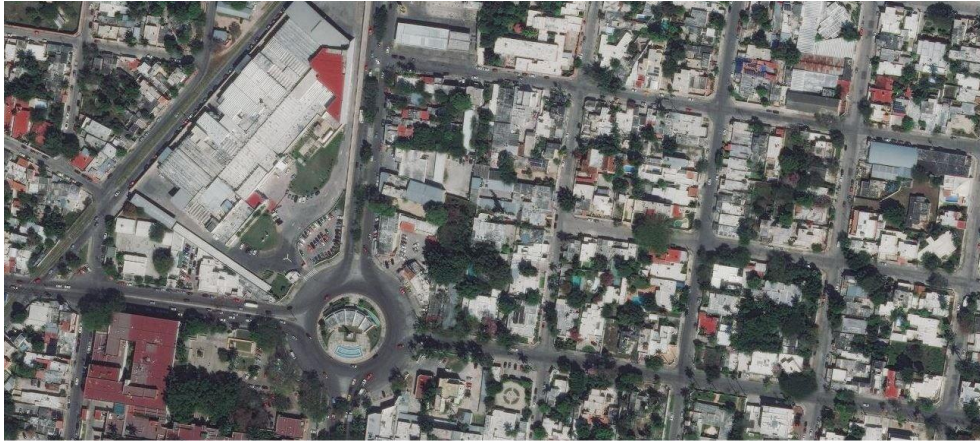
*Figura 36 - Imagen Satelital Caso 2*



*Figura 37 - Imagen Resultante Caso 2 Segmentada*

### Muestra 3

En la muestra siguiente es una parte de la ciudad, donde podemos observar casas, calles y circulación, el algoritmo de segmentación aplicado en estos y otros más casos, segmenta de manera efectiva los techos, pero tiene una peculiaridad, en los techos de color rojo, tiene lo elimina no en la mayoría dado, en otras muestras sigue el mismo patrón y de las posibles razones para la anomalía es que cuando se obtuvo el NVDI, no se especificó se utilizó la banda roja y el umbral de decantación en la mayoría toma de un techo rojo como si fuera una vegetación de tipo rojo, por otro lado los techos en la mayoría de esta sección de ciudad siguen siendo identificados por el algoritmo.



*Figura 38 - Imagen Satelital Caso 3*



*Figura 39 - Imagen Resultante Caso 3 Segmentada*

#### Caso 4

Siendo parte del aeropuerto de la ciudad de Mérida, este caso representativo, para ver el comportamiento del algoritmo, en caso donde se tenga texturas similares de asfalto representado en su forma gris, tratando de poner un caso que dificulte al algoritmo.

En la aplicación se puede ver que el techo de las salas de recepción del aeropuerto lo toma de forma correcta, pero también selección donde toman pasajeros los aviones como techos esto es porque se tiene una textura similar al techo, en realidad para el ojo humano si no estarían los aviones, se podría dudar que se techo de alguna edificación, con esto podemos ver, que a la segmentación se está haciendo correctamente, solo faltaría el acomodo de las siguientes partes.





*Figura 40 - Imagen Satelital Caso 4*



*Figura 41 - Imagen Resultante Caso 4 Segmentada*

# Capítulo III

## Clasificador con Aprendizaje de Máquina

En el capítulo se presentará el algoritmo clasificación, se comenzará con una introducción sobre los fundamentos, para contextualizar un poco sobre los métodos de *AI*, aprendizaje de máquina y demás. Luego veremos la metodología de construcción de los modelos de *machine Learning*, desde la obtención de datos hasta la parte de predicción, se presentará de la misma manera en el orden las subsecciones, primero el conjunto de datos, luego las métricas de evaluación del modelo, las características utilizadas en los modelos, el análisis y experimentación de diferentes modelos, al último se explicará a fondo el modelo que se escogió para la clasificación.

### Fundamentos

La inteligencia artificial, generalmente abreviada como *AI*, es el campo de la ciencia computacionales dedicado a la resolución de problemas cognitivos asociados comúnmente con la inteligencia humana, como el aprendizaje, la automatización de tareas cotidianas y el reconocimiento de patrones. El profesor Pedro Domingos, un gran investigador en este campo, es su libro *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, describe cinco tribus del aprendizaje virtual, las cuales son simbolistas, conexionistas, evolutivos, bayesianos y analogistas; cada una de estas tribus proviene de diferentes disciplinas, teniendo aplicaciones de modelos de *AI* revolucionando los campos de investigación como la manera de trabajo, y además de cada tribu aporta a la evolución de técnicas de *AI*.

En particular, los avances de eficiencia de la computación estadística han logrado a las tribu de bayesianos ampliar varias áreas, estas en el nombre de “aprendizaje automático”. De igual manera la computación de red han llevado a los conexionistas a crear un subcampo llamado “aprensaje profundo”. Estos son dos grandes campos que tiene en la *AI*, la cual podemos ver en la siguiente figura, como se observa en el aprendizaje automático es un subcampo de *AI* y el aprendizaje profundo es un subcampo de los dos. En rasgos generales estas técnicas se dividen en aprendizaje “supervisado”, “no supervisadas” y “reforzado” (Christianini et al, 2000). En el presente trabajo de investigación gracias a la base de datos realizada todas las técnicas utilizadas serán de aprendizaje supervisado, es decir que los datos de entrenamiento de modelos incluyen una salida deseada y etiquetada.

Hablando un poco de historia, la IA tiene el nacimiento con Alan Turing en 1950, donde crea el “Test de Turing”, luego dos años siguientes esta Arthur Samuel que escribe el primer ordenador con capaz de aprender. De manera siguiente, entra el primer invierno de la AI de 1974 a 1980, los fondos se recortaron para la investigación drásticamente. En los 80 salieron dos importante proyectos, uno de ellos es el principio de las técnicas de Natural Lagueaje Processing que es un subcampo del aprendizaje de máquinas, con el fin de que las máquinas pueden “entender” nuestro mensaje, este proyecto fue NetTalk de Terry Sejnowski. El segundo invierno del 87 al 97, esto fue un campo muerto por las lentas aportaciones del campo y reputación que duro hasta el cambio del siglo. Del 2006 en adelante es la época emergente y la adaptacion comercialmente, empezando con Geoffrey Hinton acuña el término “Deep Learning” (Aprendizaje Profundo) para explicar nuevas arquitecturas de Redes Neuronales profundas que son capaces de aprender mucho mejor modelos más planos y por último la compañía Google marcando una fecha importante en el 2016 cuando Google DeepMind vence en el juego Go (considerado uno de los juegos más complicados) afirman que el algoritmo fue capaz de realizar movimientos “creativos” que no se habían visto hasta el momento (Hawkins et al, 2016). Los avances siguen en la actualidad, las arquitecturas se vuelven más precisas gracias a la red mundial que es conocida como internet la cual facilita a enormes pasos la obtención de datos, ayudando a los modelos a mejorar.

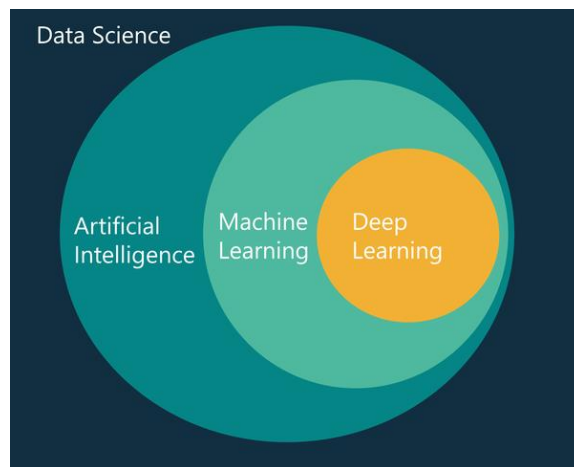


Figura 42 – Distintos campos y aplicaciones del AI, ML y DL Robert F. Dickerson

Haciendo una diferenciación entre estos conceptos, la AI es donde las maquinas son capaces de llevar a cabo las tareas de una manera considera “inteligentes” abarcando todos los campos que conlleva esto, el aprendizaje de maquina es la aplicación actual de la IA basada en que las maquinas tengan acceso a los datos y aprendan de ellos, con cierta ayuda de su definición de características que convergen el aprendizaje hacia un objetivo establecido, en términos generales se tiene los datos tratados que son las entradas del modelos, estos datos tienen salidas predeterminadas, se establecen estas arquitecturas que apoyan al entrenamiento, con un arquitectura establecida se entrena el modelo para que distinga a los siguientes datos, como en las salidas preestablecidas ( Patterson et al, 2017).

El aprendizaje profundo es la vertiente de aprendizaje automático pero la misma arquitectura extrae las características para su propio entrenamiento, es este sentido habrá diferentes capas

de datos o de procesos, representado cada una de ellas una red neuronal, individualmente sabrían hacer una cosa cumplir una función, pero colectivamente podrían cruzar diferentes habilidades para producir resultados más sofisticados y complejos. La ventaja principal es la independencia del usuario para conseguir resultados, sin necesitarlo, con los datos dado se pueden programar para que cada una de sus capas extraigan sus características necesarias como por ejemplo realizar una clasificación.

El presente trabajo se enfocará en técnicas de aprendizaje de máquina, abreviado *ml* por la traducción al inglés (*machine learning*), por dos motivos principales en el desarrollo de la investigación, se probó diferentes arquitecturas de aprendizaje profundo, pero en ninguna daba un resultado que sirvan para su exposición en el presente trabajo. La razón de ello es que el aprendizaje profundo requiere una gran cantidad de datos, en un orden de 50 mil hacia arriba (Nielsen, 2019) para que funcione. La base de datos ocupada en la presente investigación son de 400 muestras, esto porque es una base datos realizada en esta misma investigación y al ser áreas de techo factible de instalación, no se tiene el suficiente recurso, para la obtención de la base de datos, tanto en la cantidad como en la veracidad de los datos. Las arquitecturas de aprendizaje profundas dan resultados más precisos, pero están directamente relacionadas con la cantidad de datos preparados que se tienen.

## Machine Learning

Se explicará a más detalle en términos generales, como están compuesto la mayoría de los modelos, además de cómo se entrenan y evalúan sus predicciones.

Los modelos de *ml* residen su fuerza, en lugar de desarrollar un algoritmo de la manera tradicional que resuelva un problema, usamos los modelos con ejemplos etiquetados para que devuelva el algoritmo que resuelva el problema, para un mejor entendimiento ver la siguiente figura, este es una nueva forma de desarrollar soluciones que reside en los datos y en la elección de la modificación de *ml*.

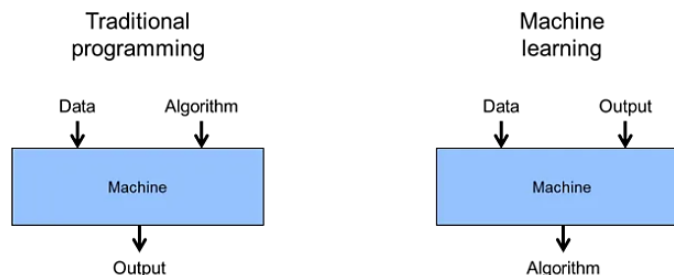


Figura 43 – Estructura de ML

Los algoritmos de *ml* obtienen sus propios cálculos según los datos que recopilan en el sistema, entre más datos mejor realizan su predicción o clasificación. Existen diferentes tipos de *ml*, para el problema en cuestión solo exploraremos el aprendizaje supervisado dado que es la manera en como estaremos manejando la información.

## Aprendisaje supervisado

Se refiere al tipo de modelos de *ml* entrenados con un conjunto de ejemplos, datos donde la salida es conocida. Los modelos aprenden de los resultados conocidos y ajustan sus parámetros interiores para adaptarse a los datos de entrada, con su separación entre sus predicciones incorrectas (Fan et al, 2005). Cuando el modelo es entrenado adecuadamente, y los parámetros internos están relacionados con los datos de entrada, el modelo podrá realizar predicciones correctas ante nuevos datos no procesados previamente, el siguiente ejemplo muestra de manera general en como es el proceso de aprendizaje del modelo.

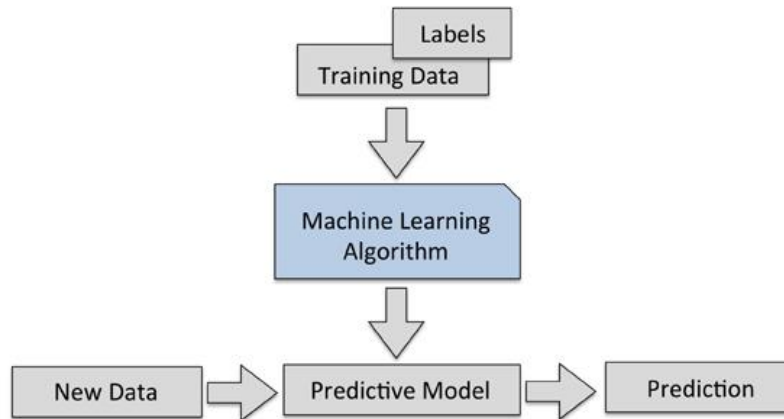


Figura 44 – Esquema General del Modelo ML

Existen dos tipos de principales según su aplicación:

### Clasificación

Es una sub-categoría de aprendizaje supervisado en la que el objetivo es predecir las clases categóricas como valores discretos, no ordenados y pertenecientes a un grupo. El ejemplo típico es la detectar spam como una clasificación binaria. También, se encuentra el clasificador multi-clase, un ejemplo detectar desde una imagen los valores de 0 a 9. Para ilustrar más, la siguiente figura muestra una clasificación binaria, que hay entre dos clases de objetos, los círculos y cruces, y están representadas por características de cada uno de los modelos que son  $X_1$  y  $X_2$ , un modelo puede encontrar las relaciones entre características de cada punto de datos y sus clases, establecer la línea divisora entre ellos. De esta manera, al ser alimentado con nuevos datos, el modelo será capaz de determinar la clase perteneciente como lo vemos con el objeto nuevo que sería la estrella:

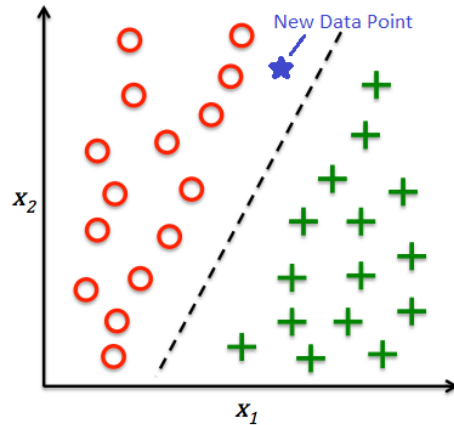


Figura 45 – Ejemplo de Modelo de Clasificación de ML

## Regresión

Estos modelos se utilizan para asignar categorías a datos sin etiquetar. En este tipo de aprendizaje tenemos un número de variables predictoras (explicativas) y una variable de respuesta (continua) se tratará de encontrar una relación entre dichas variables que nos proporcionen un resultado continuo (Chio, 2018).

Un ejemplo de regresión lineal: es de los datos  $X$  e  $Y$ , establecemos una línea recta que minimice la distancia (con el método de mínimos cuadrados) entre los puntos de muestra y la línea ajustada. Después, utilizaremos las desviaciones obtenidas en la formación de la línea para predecir nuevos datos de salida.

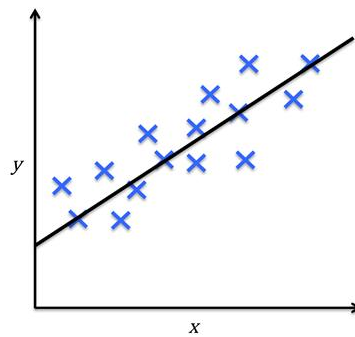


Figura 46 – Ejemplo simple de Modelo de Regresión lineal

En este proyecto de investigación se experimentó y analizó diferentes modelos de  $ml$ , cada uno de ellos tiene su propia característica particular, dado por ello no se puede generalizar los componentes, por lo cual se estará explicando en cada uno de ellos como se aplicó, y en cuales modelo nos dieron mejores resultados, estos tendrán una aplicación más afondo y detallada.

Toda la explicación se centrará para la clasificación binaria, dado que este modelo sirve para clasificar si el techo es factible o no.

## Metodología para construir modelos de ML

Aunque cada modelo tenga una arquitectura particular todos los modelos cumplen con los requisitos, con una metodología desde la obtención de datos hasta su predicción, en general son 4 pasos para desarrollar un modelo confiable, lo cuales son preprocesamiento, aprendizaje, evaluación y predicción:

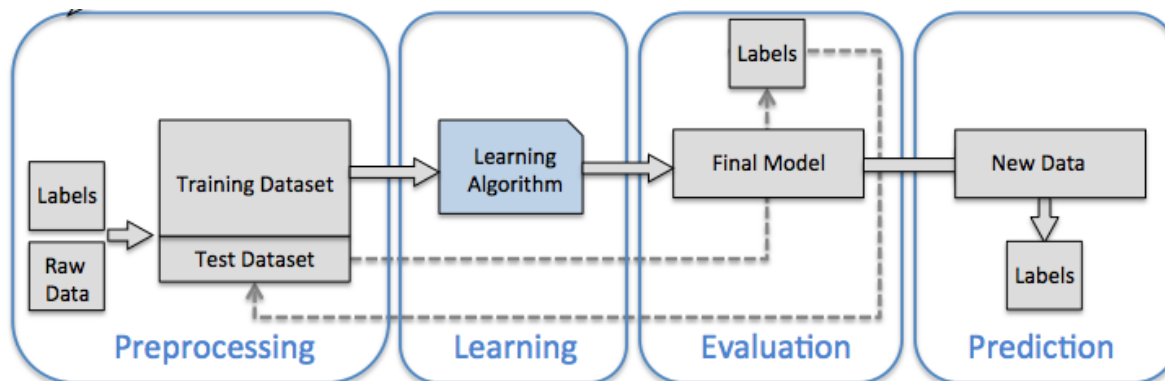


Figura 47 – Metodología Construcción de ML

### Preprocesamiento

Este paso es el más importante para cualquier modelo, dado que la forma en cómo se presentan los datos influye directamente con el funcionamiento del clasificador. En este se realizan varios puntos:

- La obtención de la base datos, la veracidad de los datos, y la verificación por el ojo humano, corroborando que sean correctos.
- La creación de base datos, etiquetar la base datos, establecer con ejemplos positivos y negativos, en la literatura se recomienda que este balanceado, 50 % positivos y 50 % negativo (Hall et al, 2018).
- Ingeniería de *feature*, la obtención de características es un proceso relacionado directamente con la retroalimentación de los resultados sobre la evaluación de los modelos, agregando, modificando y eliminando. Las características hacen una descripción más detallada que sirven de variables de entrada para los modelos de *ml*.
- Se estandariza de valores para los datos de entrada, es decir todas las características deben de estar en la misma escala, en algunos proyectos, la estandarización de los datos va desde 0 a 1, o desde -1 a 1, o estandarizan sus datos con todo el conjunto de datos, obtenido media entre la desviación estándar (Haslie et al, 2008).

### Aprendisaje

En este apartado se seleccionará diferentes algoritmos, que se integren correctamente a los datos, luego se seleccionará él tenga el mejor rendimiento. Estableceremos métricas para la evaluación de los modelos.

Por otro lado cada modelo, tiene su forma de “aprender” gracias a diferentes métodos matemáticos o estadísticos, donde se da un conjunto de datos que sirven de ejemplo y otros

que sirven para su evaluación donde se lo que predijo el modelo con lo que debió de predecir.

Evaluación:

Para cada una de las pruebas de modelos ya entrenados, se tiene un set de conjuntos de datos de evaluación, de los cuales tenemos métricas, que nos demuestran su precisión, sensibilidad, exactitud, entre otras más. Esto nos da una retroalimentación para mejorar los features, también para modificar los hiperparámetros de cada modelo experimentado, existen varias técnicas en especial para cada modelo, se explicará la teoría y aplicación en el siguiente apartado cuales se utilizaron y por qué se utilizaron.

Predicción

Luego de entrenar el modelo, hacer una evaluación de sus resultados, con nuevos datos comenzaremos a clasificar. Este apartado es cuando se integra al algoritmo general, los nuevos datos vienen de las partes de mapa que representan la ciudad, el clasificador recorre las partes de techo de la ciudad, clasificando si son factibles para la instalación de paneles fotovoltaicos o no.

## Metodología de construcción del modelo ML

En el siguiente esquema nos muestra cómo se desarrolló el algoritmo, la manera en cómo se determinó cuáles eran las características, cuales features sirvieron en la entrada de datos de los modelos de *ml*, y cómo se automatizó este proceso.

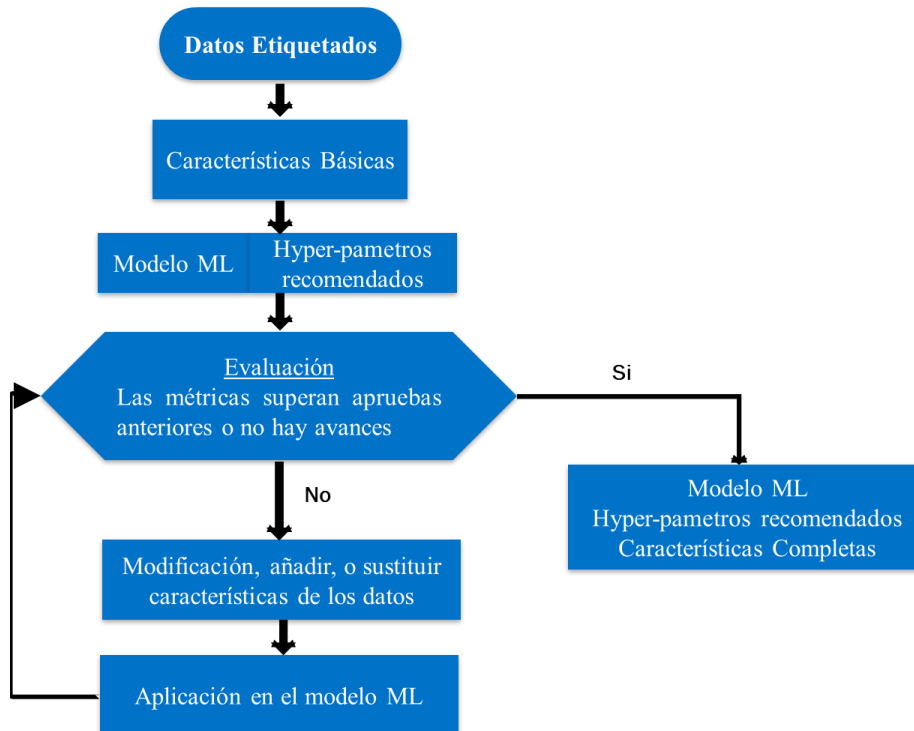


Figura 48 - Determinación de features para los modelos ML



En anterior diagrama de flujo, evalúa diferentes características, dado que en la literatura no se encuentra recomendaciones para la detección de techos sobre las características, por ello con análisis de textura y tratamiento de imágenes se estuvo probando diferentes características de las muestras en el *dataset*, con enfoque en las texturas y la geometría de píxeles, para determinar con qué conjunto de ellos darían mayor métrica en términos de clasificación.

Con las features básicas, es decir con los que se intuía que daría alguna resolución se ejecutó el modelo por primera vez, los hyper-parámetros de cada modelo fueron los que recomienda la literatura para problemas de clasificación (Mohri et al, 2012), luego se evalúa con las métricas establecidas los resultados, si son satisfactorios o no, en dado caso que se va añadiendo, modificando o descartando los features, esto se hace con todos los modelos de *ml* programados de manera automatizada cada dé vez que cambia o añade un feature el algoritmo del diagrama de flujo anterior lo aplica en todos los modelos de *ml* del stock los resultados ,guardar cada uno en una base de datos. El *testing* se desarrolló con el lenguaje de programación de Python versión 3.8 , y con el gestor de base de datos SQLite. Cuando los resultados de los modelos de la métrica no aumentan contra los ejemplos anteriores, se determinaba que ese conjunto de features es el que sirve para los modelos de *ml*. Este proceso como lo mencionado en otras investigaciones es un proceso manual, y depende la habilidad del constructor del modelo de *ml* en conjunto con el análisis de muestras (Bishop, 2006) .

Luego de tener los modelos de con los *features* completos, se debe realiza la modificación de sus hyperparametro para visualizar el crecimiento en sus métricas de clasificación o el estancamiento de ellas, de igual manera este proceso se automatizo.

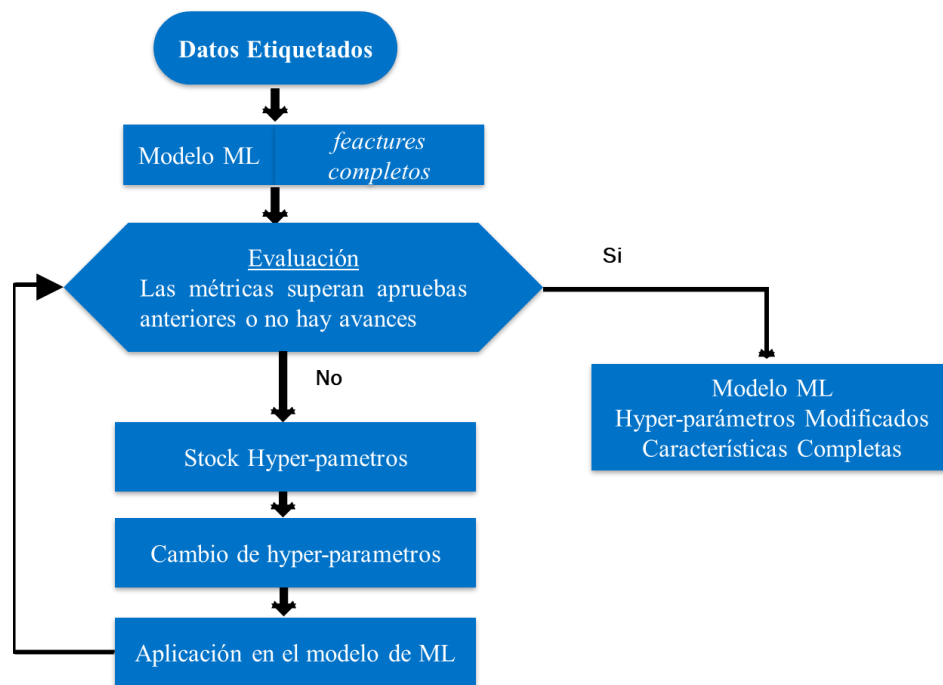


Figura 49 – Determinación de los hyper-parámetro en los modelos de *ml*

Para todos los modelos probados con los *features* completos, se creó un stock de hiperparámetros para cada modelo, los hiper-parámetros, son propiedades que se puede modificar para que el modelo se adecue de manera a la resolución del problema, es único para cada caso.

Para la evaluación de cada modelo se hacen diferentes combinaciones de cada hiperparámetro del stock, entonces el algoritmo aplica todas las combinaciones de hiper-parámetros al modelo, y luego evalúa cuál de estas combinaciones son las que dan un mayor margen al momento de clasificar, esto se realiza para cada modelo de *ml*. La automatización de este proceso, es una ventaja dado que no se necesita que el investigador encuentre la ejecución del algoritmo pero requiere un considerable tiempo de cálculo, por ello para todo el algoritmo del esquema anterior se utilizó la librería de Python la cual es CUDA<sup>2</sup>, es una herramienta que permite compilar los scripts y corran en el GPU de las computadora, con ello se puede dividir en la mitad o hasta un cuatro el tiempo de ejecución del algoritmo, lo cual fue de mucha ayuda. Todo el proceso de combinación de stock de hiper-parámetros para siete modelos de *ml*, fueron 4 días de cálculo, todo se guardó en un base datos para evaluación y análisis.

Luego de determinar los hiper-parámetros correspondientes y los *features* completos, se probó los modelos finales, el modelo que tuvo las métricas más altas contras los otros, es el que se escogió, como el algoritmo clasificador de la investigación.

## Data-set

El conjunto de datos para este modelo se trabajó con los mapas de Esri – ArcGis (para ver más de los detalles ir a la sección de obtención de imágenes satelitales), extrayendo secciones de techo, las cuales son de dos tipos, secciones de techo factible para la instalación de paneles fotovoltaicos, y secciones de techo que no sean factibles es decir el contorno del techo, techos contaminados con varios obstáculos visibles y cualquier otra muestra que no sea techo, entonces se definió nuestro conjunto de datos:

- Ejemplos positivos: secciones techo visiblemente factibles para la instalación.
- Ejemplos negativos: secciones del mapa, que muestre los techos no factibles o cualquier otro objeto.

En este conjunto de datos cada sección es de 5 pixeles de ancho por 5 pixeles de largo, esto en el tamaño real son 2.5 m por cada lado y 6.25 m<sup>2</sup>. Esta medida son las dimensiones de un techo mínimo determinado en los estándares de construcción de la mayoría de ciudades de México.

Para los ejemplos positivos de la base de datos se llevó un tiempo considerable, debido a la validación humana que se realizó para cada muestra, con cada dos pasos se verifica con el ojo humano, si se nota que la sección de techo era factible para la instalación, en dado caso que existiera alguna duda se consultaba al ayuntamiento de la ciudad de Mérida en su página web de catastro<sup>3</sup>, en ella nos muestran los planos de construcción de cada casa, validando de la mejor manera posible, esto es un proceso tardado por la verificación humana. Debido a

ello nuestra base datos cuenta con alrededor de 200 muestras entre positivas y negativas. La siguiente figura muestra una porción del conjunto de datos:

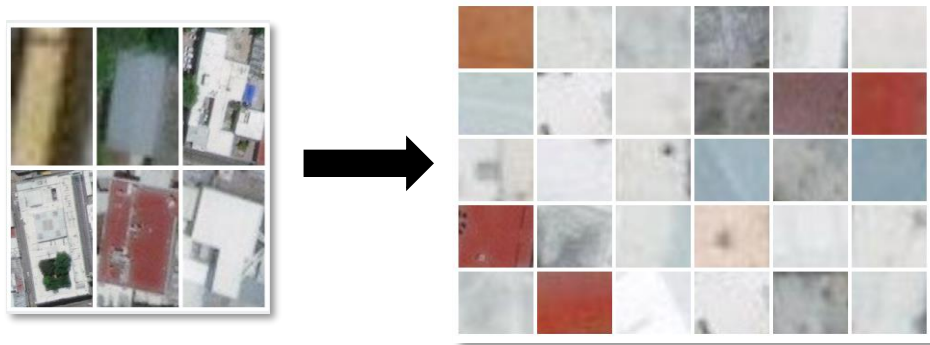


Figura 50 – Muestra Conjunto de Datos

En la anterior figura, en la izquierda se muestra los techos de las casas y en la parte izquierda se muestran las secciones de los techos los cuales son los ejemplos tanto positivos como negativos.

## Métricas

En esta sección se verá las métricas correspondientes a los modelos:

### Validación del modelo

Para la construcción de un modelo de *ml* se tiene que separar el conjunto de datos entrenamiento contra los datos de *test*, tanto los datos de validación, esto se realiza para poder tener una mejor visualización de comportamiento del modelo, existen diferentes técnicas, pero en la literatura es recomendable un modelo de clasificación utilizar el método de K-Fold Cross Validation (Bouckaert, 2003). Es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba, la siguiente figura muestra cómo es que el método K-Fold Cross Validation funciona:

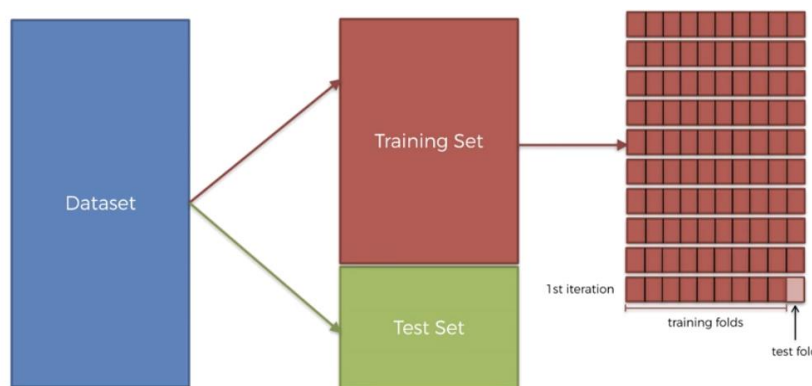


Figura 51 - Method K-Fold Cross Validation

De lo anterior todo el conjunto de datos se separa en proporciones tanto los datos de entrenamiento como los datos de prueba, ello se determina en cuantos experimentos se requiere realizar, esto se determina por la variable K. En cada experimento el conjunto de datos se va cambiando en diferentes proporciones consecuentes para cada experimento de todo el conjunto de datos. Es decir, en nuestro caso hicimos diez experimentos,  $K = 10$ , se reparte proporcionalmente e itera en los datos de prueba en diez por ciento, va de diez en diez sin repetir los datos de prueba pasados como los datos de entrenamiento, así sucesivamente, hasta terminar las combinaciones. En la siguiente figura se muestra de una manera más clara. Ejemplo para el experimento uno del proyecto, se tomó los primeros 180 ejemplos mitad positivos y negativos, y los 20 restantes fueron los datos *test*. En el siguiente experimento se toma los primeros 160 y luego los 180 a 200 para los datos de entrenamiento y de los 160 a 180 se toman los datos de prueba, de esta manera se completan los 10 experimentos tomando el 10 %, iterando para cada lado.

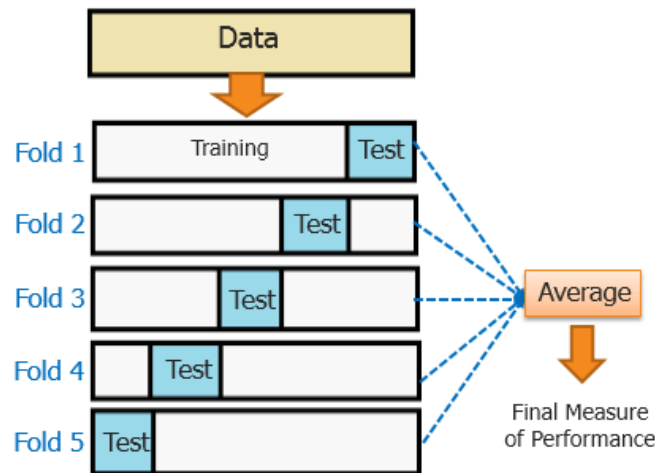


Figura 52 – Funciona K-Fold

Luego de realizar cada experimento, se tiene que determinar una métrica, que nos indique, si el clasificar binario está separando los pruebas con tanta precisión, exactitud y demas, se enfocara en varias métricas que revele correctamente el comportamiento del modelo. Es recomendable no solo para modelos de clasificación, dos clases si no para varias clases de la matriz confusión, las métricas que se pueden obtener de esta como son, exactitud, especificidad, precisión, *recall*, F1 y el modelo ROC (característica operativa del receptor), todo ellos dan una referencia detallada para ver el rendimiento del clasificador (Scholkopf et al, 2001).

### Matriz de Confusión

Una matriz de confusión es una herramienta que genera una visualización del desempeño de un algoritmo. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa la instancias en la clase real. Uno de sus grandes ventajas es que permite observar si el modelo está clasificando erróneamente las diferentes clases o los resultados de la clasificación (Kecman et al, 2005). Un ejemplo si se tiene 99 muestras

como resultado de la primera clase, pero solo 1 con la segunda clase, el clasificador tendrá un sesgo a clasificar hacia la primera clase. Es decir que clasifico todas como la primera clase, su precisión será del 99 %, esto no quiere decir un buen rendimiento, porque tiene un error de clasificación hacia la segunda clase. La siguiente figura muestra, la matriz de confusión:

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 53 – Matriz de Confusión

Las partes verticales son las observaciones, lo que realmente es la muestra, y las horizontales es lo que predice el modelo, la combinación de muestras, el comportamiento de clasificación del modelo y cada casilla significa:

- VP: verdadero positivo (TP *true positive*) es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.
- VN: verdadero negativo (TN *true negative*) es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- FN: falso negativo (FN *false negative*) es la cantidad de positivos que fueron clasificados incorrectamente como negativos. Error tipo 2.
- FP: falsos positivos (FP *false positive*) es la cantidad de negativos que fueron clasificados incorrectamente como positivos. Error tipo 1.

Si las casillas de VP y VN, de las muestras a probar son las que tiene más, significa que nuestro modelo está trabajando correctamente, pero en la práctica siempre existen variaciones y errores de tipo 1 o tipo 2 en la clasificación, por ello también tomamos otras métricas que son derivadas de la matriz de confusión para hacer un análisis completo del modelo.

#### Exactitud (*Accuracy*)

Se refiere a lo cerca que está el resultado de una medición del valor verdadero. En términos estadísticos, la exactitud está relacionada con el sesgo de estimación. También se conoce como Verdadero Positivo (o “True Positive Rate”). Se representa por la proporción entre los positivos reales predichos por el algoritmo y todos los casos positivos. En resumen, es la cantidad de predicciones positivas que fueron correctas:

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN}$$

## **Precisión**

Es la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión. Se representa por la proporción entre el número de predicciones correctas (tanto positivas como negativas) y el total de predicciones:

$$Precision = \frac{VP}{VP + FP}$$

## **Sensibilidad (Recall or Sensitivity)**

Nos indica la capacidad de nuestro estimador para discriminar los casos positivos. Es la proporción de casos positivos que fueron correctamente identificados, un ejemplo común en términos de salud, son los verdaderos positivos entre el total de enfermos, es la capacidad de poder detectar correctamente la enfermedad entre los enfermos.

$$Recall = \frac{VP}{VP + FN}$$

## **Especificidad (Specificity)**

Tasa de verdaderos negativos, se trata de los casos negativos que el algoritmo identifico correctamente, expresa cuan bien el modelo puede detectar esa clase, un ejemplo en términos de salud, seria verdaderos negativos entre el total de sanos, es la capacidad de poder identificar los casos de los pacientes entre todos los sanos

$$Specificity = \frac{VN}{VN + FP}$$

## **F1**

Combina precisión y sensibilidad en una sola métrica. Es de gran utilidad cuando la distribución de las clases es desigual, se calcula de la siguiente manera:

$$F1 = \frac{2 * precisión * recall}{precisión + recall}$$

Conforme a esta métrica tenemos varios posibles casos

- Alta precisión y alto recall: el modelo maneja perfectamente esa clase.
- Alta precisión y bajo recall: el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
- Baja precisión y alto recall: detecta bien la clase, pero también incluye muestras de otras clases.
- Baja precisión y bajo recall: El modelo no logra clasificar la clase correctamente

Todas las anteriores métricas anteriores se pueden representar en la siguiente grafica

Matriz de Confusion		Predicho			
		Negativo	Positivo		
Real	Negativo	a	b	Verdadero Negativo (True negative rate)	$a/(a+b)$
	Positivo	c	d	Exactitud	$d/(c+d)$
		Sensibilidad	Especificidad	Precisión= $(a+d)/(a+b+c+d)$	
		$d/(d+c)$	$a/(a+b)$		

Figura 54 - Matriz de Confusión Métricas

Lo anterior lo podemos representar en la siguiente figura, un ejemplo como las métricas nos permiten notar con una visualización, como se está comportando nuestro modelo:

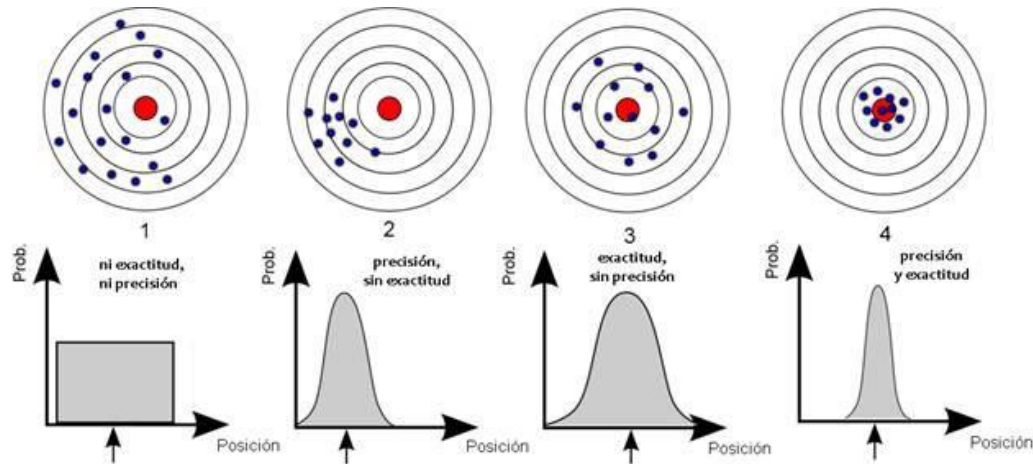


Figura 55 - Precisión y exactitud demostración (Kecman et al, 2005)

Lo anterior grafica muestra, las muestras de clasificación que sería el foco rojo se desvían de la clasificación correcta, la representación de ella está relacionada tanto con la exactitud y la precesión, esto para dar un ejemplo de la utilidad de estas métricas.

### Modelo ROC

A partir de los valores expresados en la matriz de confusión, es posible contar con una serie de medidas útiles en nuestro análisis, como la recall y la especificidad. En términos estadísticos teniendo un estimador de una variable que cuenta con un parámetro ajustable se pueden especificar sus curvas de recall y especificidad, o curvas ROC (*Receiver Operating Characteristic* o su traducción al español de Características Operativas del receptor). En la curva ROC se representa la sensibilidad del modelo frente al valor obtenido de restar la especificidad a la unidad ( $1 - \text{especificidad}$ ) (Adabi et al, 2015). Estos parámetros muestran la tasa de verdaderos positivos contra la tasa de falsos positivos, estas también son conocidos de TVP es a recall y TFP es a especificidad, en la siguiente figura mostramos la curva de ROC, mostrando el resultado de tres ejemplos de características.

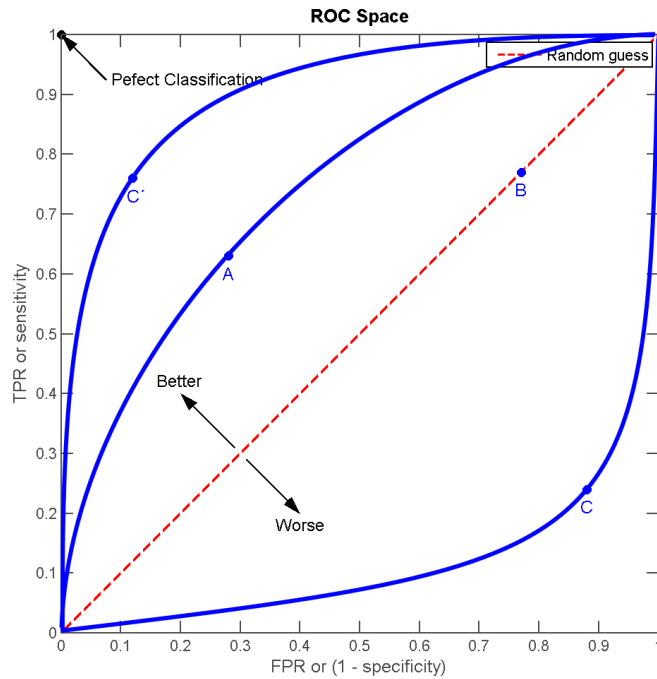


Figura 56 – Ejemplo de la curva l ROC (Zou, 2007)

La anterior figura equivale a las siguientes mediciones, de ejemplo de experimentos de un modelo cualquiera:

	A	B	C	C'
TVP	0,63	0,77	0,24	0,76
TFP	0,28	0,77	0,88	0,12
Precisión	0,68	0,50	0,18	0,82

Tabla 4 - Experimentos de la curva ROC

La curva roja en diagonal es la estimación de aleatoria, de un 50 %, que es como el azar para tener una referencia del rendimiento del clasificador.

La posición 1, que va de 0 o 100 % aparece en el grafico como la clasificación perfecta, esto se debe a que en ese punto existe una máxima sensibilidad o dicho de otra forma una recall de 100% versus TFP de 0 %, es decir que en ese punto tenemos un modelo perfecto, mientras el modelo tenga la tendencia a mejorar el rendimiento. El experimento B se encuentra sobre la línea de estimación de “azar” su precisión es de 50 %, el modelo C aparece como el peor de los tres casos con un resultado bajo, confundiendo todas tus clases.

El modelo A tiene un desempeño superior al del azar, pero no es modelo que se puede considerar de confianza para la producción. El modelo C', es el mejor modelo de desempeño, pero tambien deberían de tomar otros indicadores de la matriz de confusión. En resumir el modelo ROC nos sirve para la visualización del desempeño del clasificador conforme al recall y la especificidad dando una métrica que aporta significativamente, la decisión de escoger el mejor modelo, además de visualizar el comportamiento de los experimentos.



Las anteriores métricas nos sirven primero para poder ver la retroalimentación del algoritmo automatizado, luego realizar cambios al modelo, en cada cambio analizar que parte mejoro o el caso contrario. Esto sirvió de ayuda para la metodología de determinación de hiperparámetros y *feature*.

Todas las métricas anteriores se realizan en cada experimento del K-Fold Cross Validation, es decir en los 10 experimentos realizados se obtiene las métricas, luego se promedia los experimentos, para tener una mejor exactitud de los modelos correspondientes, de esta manera se obtiene el promedio de todos los experimentos y un análisis de más confiable.

## Features

En esta sección veremos las características finales que tienen los modelos de *ml*, las cuales generan el mejor rendimiento de clasificación para el modelo. En esta sección veremos las características teorías como sus aplicaciones, en general todas son extracciones de métodos de interpretación de imágenes satelitales. Las características se pueden clasificar en cuatro grupos, regenerativos, textura, estadísticos y de morfometria, en total son 22 características.

### Atributos Regenerativos

Este grupo de características solo se tiene uno, se inició a experimentar con varias técnicas, pero al ver que no variaba mucho la métrica del clasificador, se descartaron. Los atributos regenerativos en tratamiento de imágenes digitales muestran el cambio de pixeles entre sus vecindades, al utilizar este grupo de tratamiento primero, los cambios de pixeles nos indica, si existen obstáculos en el techo, descartando una muestra positiva del conjunto de datos.

### Módulo Gradiente 2 Dimensiones

Esta operación de detección de bordes se enfoca a encontrar los pixeles, entre dos regiones, se realizarán a través de mascarillas que se aplican sobre la imagen, en especial sobre las vecindades de cada pixel de la imagen. Para detectar los bordes entre regiones, la derivada digital es la derivada continua (Swokowski, 1998). Se define, si es necesario partir del concepto de derivada continua. Se define siempre que el limite existe:

$$\frac{df(x)}{dx} = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta) - f(x)}{\Delta}$$

El incremento ( $\Delta$ ) mínimo entre los pixeles de una imagen es un pixel, es decir  $\Delta = 1$ , la derivada digital presenta por diferencias entre pixeles, entonces se define la derivada digital como :

$$\frac{df(x)}{dx} = f(x + 1) - f(x)$$

Y con ello la derivada se puede considerar entre las diferencias entre el pixel referido y el pixel anterior:

$$\frac{df(x)}{dx} = f(x) - f(x - 1)$$

Las combinaciones de las anteriores ecuaciones son:

$$f'(x) = \frac{df(x)}{dx} = \frac{f(x+1) - f(x-1)}{2}$$

Para una imagen representada por  $f(x, y)$ , las derivadas se definen en las direcciones horizontales y verticales:

$(x-1, y-1)$	$(x-1, y)$	$(x-1, y+1)$
$(x, y-1)$	$(x, y)$	$(x, y+1)$
$(x+1, y-1)$	$(x+1, y)$	$(x+1, y+1)$

Tabla 5 - vecindad  $f(x, y)$ , al centro se sitúa el pixel con posición  $(x, y)$

Se puede definir la derivada parcial horizontal y vertical de la siguiente manera:

$$f_x = \frac{\partial f(x, y)}{\partial x}$$

$$f_y = \frac{\partial f(x, y)}{\partial y}$$

La derivada digital sobre regiones homogéneas, las regiones que tienen niveles de gris constante, el resultado será cero, opuestamente será distinto de cero en áreas discontinuidades en intensidad. Como se ha establecido, los objetos en una imagen son regiones homogéneas en intensidad, los bordes al contrario son regiones en donde la intensidad cambia.

Las mascarillas utilizadas en la operación de detección de bordes están basadas en operadores de gradientes fundamentados en el concepto de derivada digital, el gradiente de una función  $f(x, y)$  se define como el siguiente vector transpuesto:

$$\nabla f(x, y) = [G_x, G_y]^t = \left[ \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right]$$

Se puede definir la magnitud y el ángulo del gradiente de la función. En el caso de tratamiento de imágenes, la magnitud de gradiente representa el cambio máximo por distancia unitaria, el nivel de gris dentro de la imagen unitaria (Urcid, 1999).

Con las muestras de las entradas de datos del modelo se obtiene el gradiente, donde la nueva matriz es el cambio de intensidad, obtenemos la moda tanto de manera horizontal como vertical en la nueva matriz, es una operación estadística porque el gradiente más significativo que hubo en las dos direcciones, este número, nos revela cuantos cambios de intensidades de grises hubo en la muestra.

### Atributos de Textura

En el capítulo 2 se vio una introducción sobre textura, los atributos de que se obtienen de la matriz de co-ocurrencia, los cuales son los que tienen más peso al momento del aporte de la métrica del clasificador, porque muestran la homogeneidad o la heterogeneidad de la textura.

Por ello primero se explicará cómo se obtiene, luego poder ver el funcionamiento de la matriz sobre todo la textura.

La matriz de co-ocurrencia o GLCM (grey level co- occurrence matrix) describe la frecuencia de un nivel de gris que aparece en una relación espacial específica con otro valor de gris, dentro de una ventana predeterminada. En otras palabras, un resumen es la forma en que los valores de los píxeles ocurren uno al lado de otro en una pequeña ventana.

Se realizan una serie de pasos que llevando a la interpretación de imágenes de textura:

- a) Tamaño de la ventana, que está relacionada con la cantidad de niveles de grises de la imagen de entrada.
- b) Componente espacial (la distancia entre píxeles y el ángulo para el cálculo de la matriz de co-ocurrencia).
- c) Cálculo de matriz simétrica.
- d) Cálculo de matriz de probabilidad.
- e) Cálculo de imágenes de textura.

El tamaño de la ventana debe tratar de ser cuadrado o autocompletarlo con los píxeles vecinos, esto en la parte de los bordes o si es número para la ventana. El resultado del cálculo es un único valor que representa la ventana completa, el cual se encuentra en el lugar del píxel central. Luego, se mueve un píxel y el cálculo se repite para la nueva matriz de co-ocurrencia colocando este nuevo valor en la posición central de esta ventana. Se construye la nueva imagen con valores de relación entre píxeles. La siguiente figura indica cómo se conectan los valores:

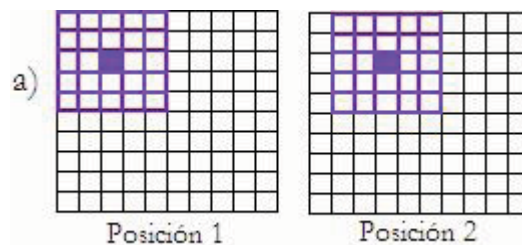


Figura 57 – Imagen de ventana móvil de 5 x 5, en dos posiciones, con el píxel centra de posición

El tamaño relativo de la ventana y de los objetos en la imagen determina la utilidad de esta medida para la clasificación. Nuestro el tamaño de la ventana es de 3 píxeles. La relación espacial entre el píxel de referencia y su vecino puede ser en cualquiera de las 8 direcciones (N, S, E, O y las 4 diagonales). Cuando se habla de una relación espacialmente “invariante” se eligen las cuatro direcciones N, NE, E y SE y se promedian. Se expresa de igual manera respectivamente como  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$  como muestra la figura:

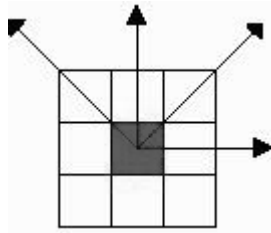


Figura 58 – Píxel y las direcciones que toma

La matriz de co-ocurrencia es una tabla que indica cómo están relacionadas las distintas combinaciones de los valores de brillo de los píxeles (niveles de grises) que ocurren en la imagen por lo que la matriz de co-ocurrencia tendrá dimensiones de  $N \times N$  si existen  $N$  niveles de grises. Para una imagen de 256 niveles de grises se tendrá una matriz de co-ocurrencia de  $256 \times 256$  para cada dirección y ventana. Se utiliza el primer ejemplo negativo la parte del píxel es un borde del techo, una sección para ejemplificar el funcionamiento de matriz de co-ocurrencia dado que si se vería toda una muestra compleja tampoco sería relevante solo alargaría los cálculos. En la siguiente figura es una muestra de cuatro niveles de grises (0, 1, 2, 3) por lo cual se tendrá una matriz de co-ocurrencia de  $4 \times 4$ :

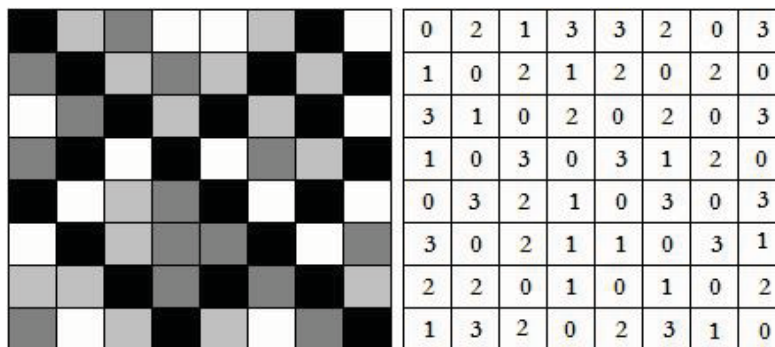


Figura 59 - Muestra de matriz de co-ocurrencia de 4 píxeles

La matriz de co-ocurrencia considera la relación espacial entre dos píxeles, llamados píxeles de referencia y píxel vecino. Por ejemplo, si el píxel vecino que está a la derecha de cada píxel de referencia, este se expresa como (1,0): 1 será el píxel en la dirección  $x$  y 0 será un píxel en la dirección  $y$ . Así, cada píxel en la ventana se convierte en el píxel de referencia empezando por esquina superior izquierda y finalizando esquina inferior de derecha. Se observa que los píxeles ubicados en el margen derecho de la anterior no tienen vecino a la derecha por lo tanto no son usados en el procesamiento por lo tanto se crean las relaciones entre píxeles:

- (1,0) es igual a  $0^\circ$  un píxel a la derecha.
- (1,1) es igual a  $45^\circ$  un píxel a la derecha y un píxel arriba.
- (0,1) es igual a  $90^\circ$  un píxel arriba.
- (-1,1) es igual a  $135^\circ$  un píxel a la izquierda y un píxel arriba.

Tomando en cuenta la dirección (1,0) mencionada anteriormente y que el píxel de referencia estará inmediatamente después de su píxel vecino, las posibles combinaciones entre los cuatro niveles de grises están mostrados en la siguiente figura:

Pixel Vecino Pixel Referencia		0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)	
1	(1,0)	(1,1)	(1,2)	(1,3)	
2	(2,0)	(2,1)	(2,2)	(2,3)	
3	(3,0)	(3,1)	(3,2)	(3,3)	

Tabla 5 - Combinaciones de cuatro pixeles

Entonces la matriz se puede interpretar como, la primera celda debe ser llenada con la cantidad de veces que ocurre la combinación (0,0), en otras palabras, cuantas veces en el área de la ventana un píxel con valor gris igual a 0 (píxel vecino), situado a la derecha de otro píxel con valor 0 (píxel de referencia). Por lo tanto, existen diferentes matrices de co-ocurrencia para cada relación espacial según se considere el venció de arriba o al lado o en diagonal. La siguiente figura muestra el resultado de la figura anterior:

0	2	8	8
8	1	2	2
8	4	1	1
3	4	3	1

Tabla 6 – Resultado de la relación entre pixeles

La matriz mostrada en la figura anterior tiene cada píxel vecino en su respectiva derecha. Si el cálculo se realiza solo de este modo, usando la dirección (1,0), entonces el número de veces que aparece la combinación (2,3) no es el mismo que la combinación (3,2), por lo tanto, la matriz no es simétrica. La simetría es necesaria para el cálculo de texturas. De la anterior figura se obtiene una simetría mostrada en la siguiente tabla:

0	10	16	11
10	2	6	6
16	6	2	4
11	6	4	2

Tabla 7 - Matriz Simétrica de co-ocurrencia

Una vez obtenida la matriz simétrica a través de su matriz transpuesta, el paso siguiente es expresar esta matriz como probabilidad

0	0.089	0.142	0.098
0.089	0.017	0.053	0.053
0.142	0.053	0.017	0.035
0.098	0.053	0.035	0.017

Tabla 8 - Matriz de probabilidad

En general, cuanto mayor es el número de la diagonal en la matriz de co-ocurrencia, más homogénea es la textura en esa parte de la imagen que está siendo analizada. Entonces, los

elementos de la diagonal representan pares de píxeles que no tienen diferencias en su nivel de gris. Si estos elementos tienen altas probabilidades, la probabilidad se refiere a la coincidencia del valor de la vecindad en conjunto de píxeles de la ventana completa, entonces la imagen no muestra mucho contraste, la mayoría de los píxeles son idénticos a sus vecinos, por lo tanto, esto le indicaría al modelo que sería un ejemplo positivo, pero como pasa lo contrario le indica que es un ejemplo negativo. Sumando los valores de la diagonal se obtiene la probabilidad de que un píxel tenga el mismo nivel de gris que su vecino.

Además, las líneas paralelas a la diagonal separadas una celda, representan los pares de píxeles con una diferencia de 1 nivel de gris, por lo que a medida que se aleja de la diagonal la diferencia entre los niveles de grises es mayor. Sumando los valores de estas diagonales paralelas se obtiene la probabilidad de los cuatro tipos de intensidades de gris con los vecinos.

Este parte como generador de los atributos que emitimos de la matriz para todos los ejemplos:

### **Homogeneidad**

Esta característica nos devuelve un escalar, donde una imagen resulta homogénea si los valores diagonales principal de la matriz de co-ocurrencia son altos, esto porque al observar la ecuación de homogeneidad los valores de probabilidad son mayores en la diagonal principal y su peso cae exponencialmente al alejarse de la diagonal:

$$Homogeneidad = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

Donde  $P$  es el píxel ubicado en las coordenadas  $i$  y  $j$  de la matriz mencionada anteriormente.

### **Contraste**

Se tendrá un valor alto si los valores altos están concentrados lejos de la diagonal principal y el peso de la probabilidad aumenta, pero en forma cuadrática, es decir es lo opuesto de la homogeneidad:

$$\sum_{i,j=0}^{N-1} P_{i,j} * (i-j)^2$$

### **Energía o Uniformidad**

Los valores de la matriz de co-ocurrencia son de probabilidades por lo que la ecuación de energía, la característica que nos arroja es realizar la maximización de los valores grandes y minimizar los valores más pequeños.

$$\sum_{i,j}^{N-1} P_{i,j}^2$$

### **Entropía**

Esta característica es la encargada de medir la aleatoriedad de los píxeles en la matriz de co-ocurrencia (Shannon, 1948). La medida presenta la información proporcional (continua). Es decir, un pequeño cambio de las probabilidades de aparición de los elementos de la señal

debe cambiar poco la entropía. Por lo que se espera que una imagen suave tenga una baja entropía y una alta entropía será para imágenes toscas, un techo con varios obstáculos.

En dado caso que todos que den una imagen uniforme sin ningún obstáculo, la entropía será máxima.

$$\sum_{i,j}^{N-1} -P_{i,j} \ln(P_{i,j})$$

Asumiendo que será  $0 * \ln(0) = 0$

### Correlación

Devuelve una medida de la correlación de un pixel con su vecino en toda la imagen, el rango va desde un rango de [-1, 1]:

$$\sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)p(i,j)}{\sigma_1 \sigma_2}$$

La correlación va desde 1 o -1, para una imagen perfectamente positiva o negativa correlacionada, la correlación es 0 cuando una imagen es constante.

### Atributos Estadísticos

En este apartado se sacarán los atributos estadísticos de los datos de secciones de techo, todos los ejemplos son modelos RGB con una profundidad de tres matrices, para estas características se transformaron primero en muestras en escalas de gris transformándoles en una matriz de  $n \times m$ . A continuación, explicamos a más detalle cómo se obtuvieron las características estadísticas:

### Desviación estándar

Esta técnica de estadística se utiliza para cuantificar la variación o dispersión del conjunto de datos, en términos del ejemplo de la sección de la imagen digital, podemos ver como una dispersión mayor es notario de un ejemplo negativo, la siguiente ecuación nos ayuda a determinar la medida:

$$\sigma = \sqrt{\frac{\sum_1^N (x_i - \bar{x})^2}{N}}$$

### Coefficiente de correlación

El coeficiente de correlación, en términos generales es la relación que tienen uno con la otra variable, de esta manera el crecimiento de cada uno, tenga significado hacia el otro, esta característica nos indica que la correlación de una imagen contaminada sea desigual.

El coeficiente de correlación de dos variables aleatorias es una medida de su dependencia lineal, la correlación va ser la matriz de ejemplos positivos y la matriz transpuesta del mismo ejemplo. Si cada variable tiene  $N$  observaciones escalares, entonces el coeficiente de correlación se define como:

$$\rho(A, B) = \frac{1}{N-1} \sum_{i=1}^N \left( \frac{A_i - \mu_A}{\sigma_A} \right) \left( \frac{B_i - \mu_B}{\sigma_B} \right)$$

Donde  $\mu_A$  y  $\sigma_B$ , son la media y la desviación estándar de A, respectivamente también para la desviación de B. Alternativamente, se puede definir el coeficiente de correlación en términos de covarianza de A y B, recordando donde A es la matriz en escala de grises del ejemplo ya sea positivo o negativo.

$$\rho(A, B) = \frac{cov(A, B)}{\sigma_A \sigma_b}$$

La matriz de coeficientes de correlación de dos variables aleatorias es la matriz de coeficientes de correlación para cada combinación de variables por pares (Sezgin, 2014).

### Media

En esta característica se especifica la media de las primeras columnas creando una matriz de una sola fila, luego sacando la media a dicha fila, y de esta manera obtenemos un escalar, que nos indica el promedio de las intensidades de grises:

$$\mu = \frac{1}{N} \sum_{i=1}^N A_i$$

Donde N es el la cantidad de pixeles,  $A_i$  es el i-esimo pixel del ejemplo para la entrada del modelo.

### Desviación absoluta media

Esta medida se utiliza en lugar de la desviación media cuando es necesario que los valores extremos afecten menos al valor de la desviación. Debido a que los valores extremos afectan menos a la mediana o a la media. El valor extremo para los ejemplos de entrenamiento, son las diferentes intensidades de grises afectados conjunto de datos del ejemplo:

$$DMA = \frac{\sum |x_i - \mu|}{N}$$

Donde la  $\mu$  es la media de la población, el  $x_i$  es el valor del i-ésimo pixel o dato y el  $N$  es la cantidad de pixeles de nuestro dato.

### Atributos Cuantitativos Morfometria

En esta sección de los atributos se toma los métodos de morfometria que son las propiedades geométricas que tiene una imagen digital, estas técnicas permiten la comprensión de la forma de un objeto dentro de la imagen digital, convergiendo a la interpretación de la imagen. Se obtuvieron varias características que nos muestran los datos:

#### Área

Se transforma a una imagen binaria, entonces el área es el número de real de pixeles que se tiene en la imagen, excluyendo el fondo, para interpretación de nuestro contexto, son cuantos obstáculos de la imagen se encuentran binarizados.



## Cuadro delimitador

Es el rectángulo más pequeño que contiene a la región, regresada como una 1-por-vector ( $2 * Q$ ), donde  $k$  es el número de dimensiones de la imagen. Entonces se expresan que los primeros  $Q$  son las coordenadas de la esquina mínima de la caja, los segundo elementos de  $Q$  son el tamaño del cuadro a lo largo de cada dimensión. Un ejemplo sería la siguiente figura:

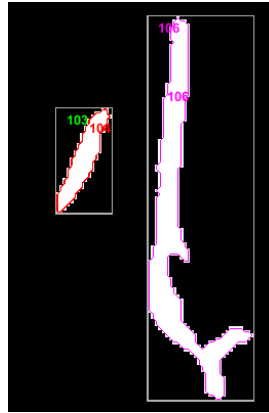


Figura 60 - Ejemplo de cuadro delimitador

El proceso se hace de iteración con operadores de frontera, esto nos devuelve cuantos cuadros delimitadores hay en nuestra imagen lo que nos dice cuántos elementos de obstáculos se tienen.

## Centroide

El Centroide nos indica el centro de masa de la región, devuelto como un 1 por  $Q$ . El primer elemento del Centroide es la coordenada horizontal, o coordenada  $x$  del centro de masa. El segundo elemento es la coordenada vertical o coordenada  $y$ . Todos los demás elementos de centroide están en orden de su dimensión. En la siguiente figura se muestra el centroide y el cuadro delimitador para una región no contigua, la región tiene los píxeles blancos, el cuadro verde en el delimitador, donde el punto rojo es el centroide.

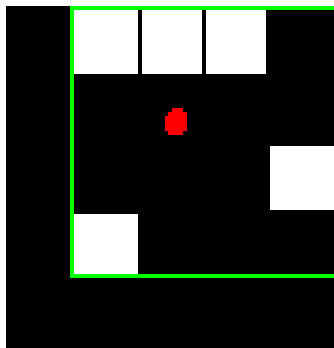


Figura 61 - Ejemplo de Centroide

Lo que nos devuelve es un escalar dándonos cuantos centroides se tienen en cada región obteniendo la media en algunas regiones dentro de los datos también puede verse más, de esta manera obtenemos el número de centroides.

## **Longitud de Eje mayor y menor**

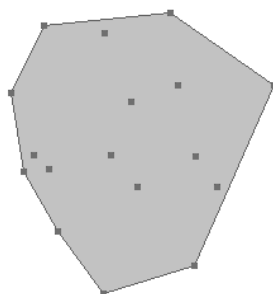
Longitud en píxeles del eje mayor de la elipse tienen los mismos momentos de central segundo normalizados como la región, es decir se binarizada la imagen y las regiones dentro de las muestras, es decir los objetos se toman de esquinas a esquina, donde los objetos dentro de la imagen se obtiene el mayor siendo un escalar, de igual manera se realiza para el eje menor, pero en este caso el objeto se determine más pequeño. Esta medida indica que tan extenso este obstáculo en el techo.

## **Excentricidad**

Esta medida nos muestra la relación entre la distancia del foco de la elipse y la longitud de su eje principal, al referirnos a elipse es la distancia más lejana entre los puntos. El valor esta entre 0 y 1, una elipse cuya excentricidad es 0 es en realidad un círculo, mientras que una elipse cuando su excentricidad es 1 es un segmento de línea. Se debe de recalcar que la elipse es la separación del objeto dentro del ejemplo. La medida no indica la forma del objeto dentro de la imagen.

## **Área Convexa**

Esta característica nos especifica el casco convexo, son todos los píxeles dentro del casco, de los objetos de la ventana a interpretar. La imagen, tiene el tamaño del cuadro delimitador de la región, para los píxeles que atraviesa el límite del caso, se determine el área de la imagen. Como se ve en la siguiente figura:



*Figura 62 - Ejemplo de Área Convexa*

Todos los píxeles de esta área se encuentran en los diferentes objetos dentro de la imagen, para nuestra medida se denota un escalar, que nos da el número de píxeles dentro del área convexa de una imagen en binarizada, nos da un vector con varios números conformes más objetos se tenga, se saca el promedio, esta medida indica la cantidad y el tamaño de los obstáculos (objetos) en el techo.

## **Área Rellena**

El número de píxeles dentro de una imagen rellena, es decir, dentro de los objetos binarizados se diferencian del fondo (los píxeles negros de la imagen), los objetos que estén continuos se cuenta el número de píxeles, dentro de ellos, e igual que el anterior entre más objetos

separados en la imagen mayor áreas rellenas, por lo tanto, se saca la media de la cantidad de estos para la relación de los objetos.

### Numero de Euler

La constante matemática es uno de los numero irracionales más importante, para la característica de tratamiento de imágenes digitales, son el número de objetos en la región menos el número de agujeros, es decir si en un objeto no se encuentra totalmente relleno se tiene un agujero por lo tanto se comienza a acrecentar el número de Euler.

La medida nos indica como el obstáculo en el techo, o el no techo factible tiene dentro de una figura discontinua.

### Solidez

Es la proporción de los pixeles en el casco convexo en la región, entre el área medida es determinada de la siguiente manera:

$$\frac{Area}{Area Convexa}$$

### Grado

Es la proporción de pixeles de la región en los pixeles en total, el cuadro más pequeño dentro de la región, en decir, calcula el área divide por el área del cuadro delimitador.

### Perímetro

La distancia alrededor de la frontera de la región, calcula el perímetro de la distancia entre cada par adyacente de pixeles alrededor de la frontera de la región. Para cada muestra nos da una frontera de la región.

### Intensidad Promedio

Nos regresa la intensidad de la región, el promedio de cada objeto que se tenga en la muestra, entre mayor sea el número se tiene más objetos con intensidades de incremento diferentes.

En resumen, se explicó la teoría y el funcionamiento de todas las características de entrada al modelo de *ml*, a continuación, se observa una tabla de resumen de todas las características vista anteriormente. Estas características, ingresarían al modelo para cada dato de ejemplo positivo y negativo del modelo, en modo de visualización:

<i>Features</i>	Atributos regenerativos	Gradiente 2 D
	Atributos de textura	Homogeneidad
		Contraste
		Energía
		Entropía
	Atributos estadísticos	Correlación
		Desviación estándar
		Coficiente de correlación

		Media
		Desviación absoluta media
	Atributos cuantitativos	Área
		Cuadro delimitador
		Centroide
		Longitud del eje mayor
		Longitud del eje menor
		Excentricidad
		Área convexa
		Área rellena
		Numero de Euler
		Solidez
		Grado
		Perímetro
		Intensidad promedio

Tabla 9 – Resumen de Features

## Análisis y experimentación de modelos *ml*

En este apartado veremos los modelos de *ml* que se experimentó y analizo con la metodología, estos modelos muestran el comportamiento de los datos en modelos diferentes entre cada uno, todos son modelos clasificadores, cada uno de ellos se vera de manera general sin detallar, el único que se explicará a fondo será el modelo que dio el mejor rendimiento en el siguiente capítulo.

### Regresión logística

Es un modelo estadístico, se utiliza cuando estamos interesados en pronosticar la probabilidad de que ocurra o no un suceso determinado, en este caso la probabilidad de que sea un techo factible o no.

Los modelos de regresión estudian si una variable binomial depende, o no, de otra u otras variables (no necesariamente binomiales). Si una variable binomial de parámetro  $p$  es independiente de otra variable  $X$ , se cumple  $p = \frac{p}{X}$ , por consiguiente, un modelo de regresión es una función de  $p$  en  $X$  a través del coeficiente de  $X$  permite demostrar la relación anterior. Por lo tanto, es una extensión de los modelos de regresión lineal, con la particularidad de que el dominio de salida de la función está acotado al intervalo  $[0,1]$  y que el procedimiento de estimación, en lugar de mínimos cuadrados, utiliza el procedimiento de estimación máximo-verosímil. El análisis de regresión logística sólo precisa del principio de monotonía, es decir, si el suceso  $A$  determinara que es un techo factible, deberá de ocurrir:

$$x_i > x_j \iff P\left(\frac{A}{x_i}\right) > P\left(\frac{A}{x_j}\right)$$

Entonces, tenemos que poner su función de costo la cual nos va dar las iteraciones de cómo se van actualizando sus pesos, las 22 características que seleccionamos previamente, con la función de costo determinara nuestro polinomio es:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}]^2$$

Donde  $\theta$  es el peso que tiene las características, de cada entrenamiento para  $x$   $i$ -ésima son los datos de entrenamiento  $x$  y  $y$  son los datos de salida la diferencia tiene que ser menor para la retroalimentación de los pesos.

Para su optimización se realiza con el gradiente estocástico, que es la derivada parcial de nuestra función hasta encontrar su mínimo local, lo cual converge en el gradiente estocástico. Entonces para la determinación de los pesos del polinomio tenemos que, optimizar el pseudocódigo siguiente:

```

Bucle {
    Para cada  $i = 1$  a  $m$  {
 $\theta_j := \theta_j + \alpha [y^i - h_{\theta}(x^i)] x_j^i$       (para cada  $j$ )
    }
}

```

Donde  $\theta_j$  son los pesos, el  $\alpha$  es un hiperparámetro, una constante de saltos de cada iteración  $m$  son las iteraciones que se tiene hasta actualizar los pesos es un hiperparámetro de igual manera, las  $j$  son las 22 características. Para la predicción se pone un umbral que si es mayor al 0.70 entonces se toma como una muestra correctamente clasificada.

En este modelo, los mejores hiperparámetros que se pudieron obtener fueron:

- Polinomio de grado 1
- $M = 1500$  iteraciones
- $\alpha$  de 0.01

La siguiente figura muestra como el error de  $J(\theta)$  va disminuyendo conforme las iteraciones donde el error va en decaída, del modelo:

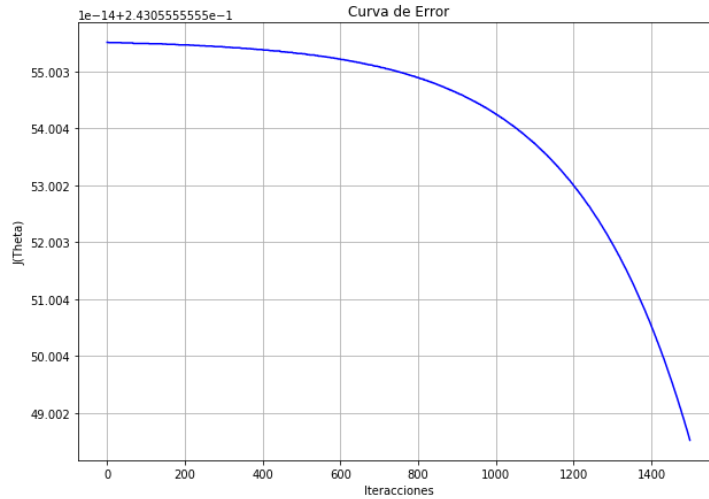


Figura 63 - Curva de Error Regresión Logística

Y con el método de *K-fold validation*, el promedio de los diez experimentos, es el siguiente, para 20 muestras de prueba:

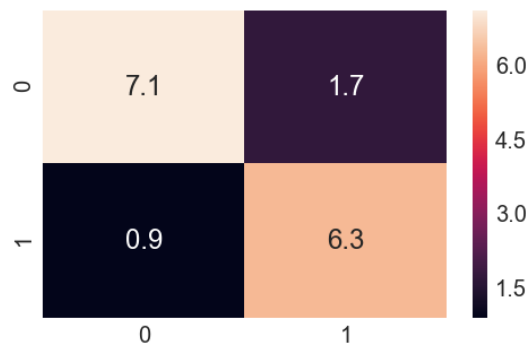


Figura 64 – Matriz de confusión de K fold de regresión logística

Se puede observar que una tendencia pequeña clasificar falsos positivos, las métricas que obtuvimos de la anterior matriz fueron:

Recall	0.8841
Specificity	0.8130
Accuracy	0.8375
Precision	0.7875
F1	0.8273

Figura 65 – Métricas de regresión logística

Es un modelo que tiene el defecto de los fasos positivos ligeramente por ello en la métrica de *precisión* tiene una disminución, donde tiene un mayor rendimiento es en el recall o sensibilidad, que nos dice que clasifica bien los positivos.

En la siguiente figura vemos las curvas de los diez experimentos del modelo ROC:

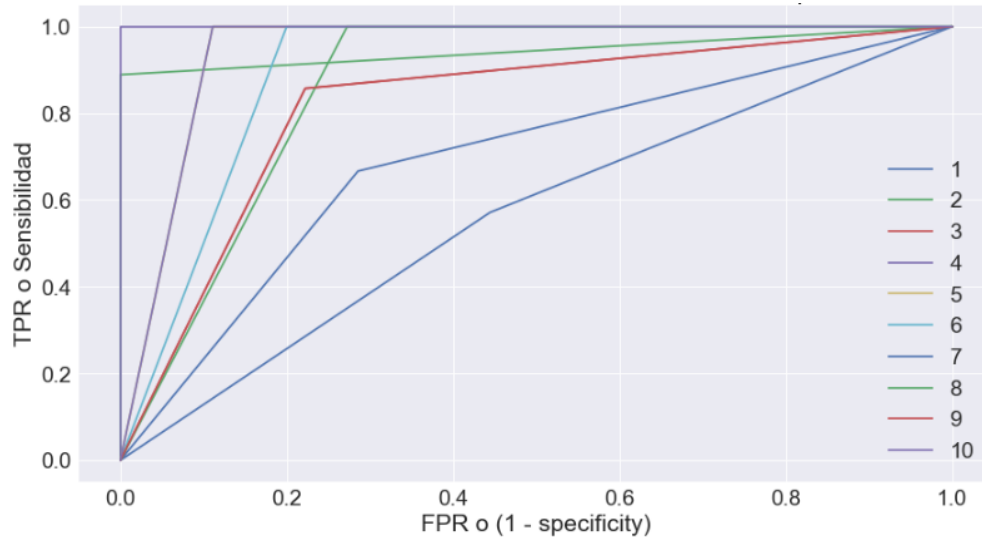


Figura 66 – Modelo ROC Regresión Logística

Los promedios de las curvas nos indican **84 %**, para recalcar este número, este el indicador con más peso tomar, que influye la decisión el cual proporciona la relación de clasificar los positivos y los negativos.

### Análisis Discriminante Gaussiano

Es un método de variables cualitativas (las características) en la que dos grupos con sus observaciones se clasifican en función de características. Haciendo uso del teorema de Bayes, se estima la probabilidad de que una observación, dado un entrenamiento de valores predictores, pertenezca a cada una de las clases de la variable cualitativas,  $P(Y = k | X = x)$ , finalmente se le asigna la observación a cada clase  $K$  para que la probabilidad sea mayor.

En este modelo, asume que  $p(x/y)$  es distribuido acorde a la distribución multi-variable normal en  $n$  dimensiones, es decir que nuestra matriz de datos tiene 22 características, las cuales son las filas y la última fila son ejemplos positivos y negativos, las columnas son los datos de entrenamiento, entonces tenemos una matriz de 23 filas con 180 columnas. Con este conjunto de datos se aplica la siguiente formula y obtenemos:

$$p(x, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu)^t \Sigma^{-1}(x - \mu)\right)$$

Donde

$$\mu \in \mathbb{R}^n \text{ es la media del vector}$$

Y

$$\Sigma \in \mathbb{R}^{n \times n} \text{ es la matriz de covarizansa y } \Sigma \geq 0$$

Con ello sacamos las características donde  $n$  es la dimensión de la matriz, de esta manera sacamos la probabilidad para cada caso conforme a todos los datos, es decir se calculó un

ejemplo para toda la matriz de características y con respectivas muestras positivas y negativas.

En la siguiente figura se muestra el resultado del promedio de los 10 experimentos, de la matriz de confusión, de los 20 datos de prueba:

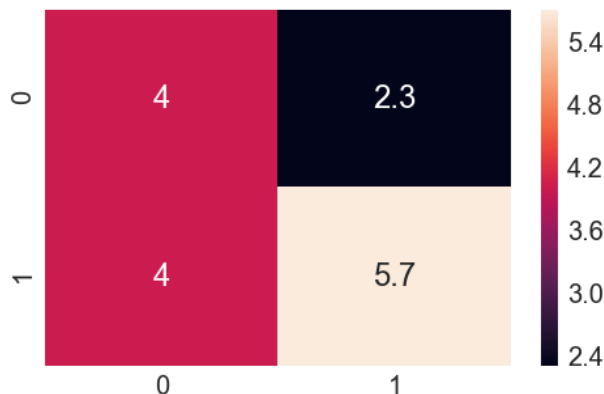


Figura 67 – Matriz de confusión DGA

Como se puede ver en la anterior figura, se tiene varias clasificaciones erróneas tanto en los positivos como en los negativos, los que tuvieron mejores datos fueron verdaderos positivos, pero aun así la red está debajo del promedio requerido, y se puede observar también en sus métricas:

Recall	0.5955
Specificity	0.6246
Accuracy	0.6062
Precision	0.7125
F1	0.6455

Figura 68 – Métricas de DGA

Aunque por la literatura nos dice que, si pasa mayor del 50 %, significa que el modelo es un clasificador regular, estos modelos se presentaron por estos argumentos (Hastie, 2008). Uno de los defectos principales que vemos en sus métricas es que confunde los ejemplos negativos, con lo cual no clasifica correctamente, por lo tanto, el F1, la exactitud y demás se tiene menos del 70%. Continuando con este compartimiento su curva ROC del modelo también presenta este defecto:



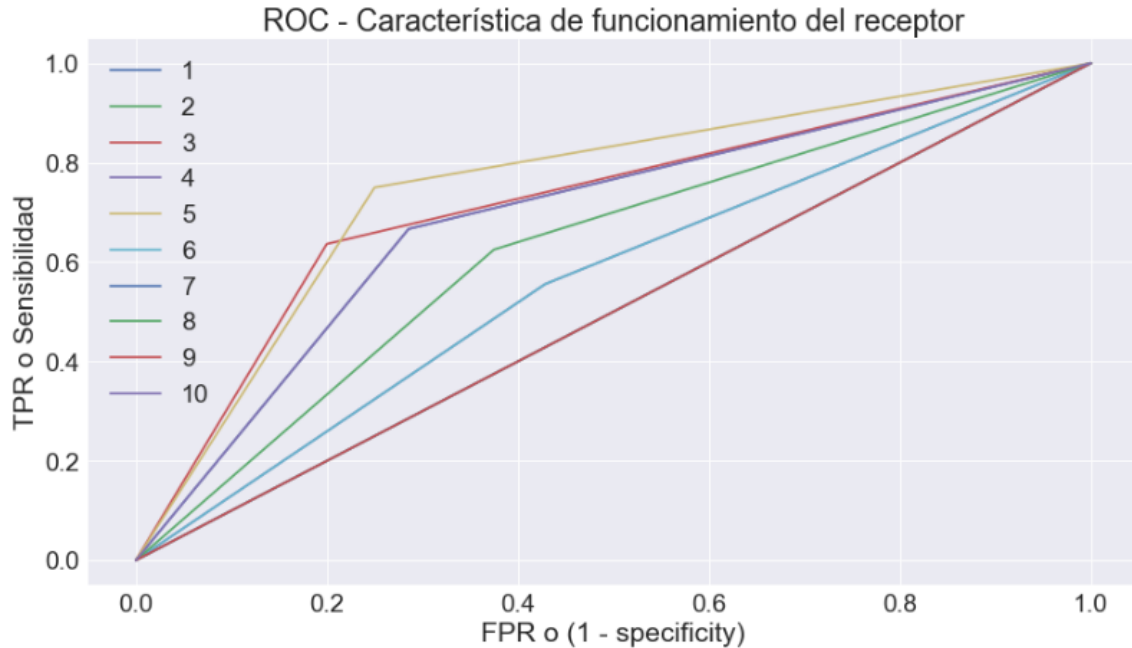


Figura 69 – Modelo ROC de DGA

Como se puede ver en la anterior gráfica, en el quinto experimento es donde casi el 80 %, en todos los demás estuvieron debajo, este modelo es el eje de FPR no indica que clasifica mejor los positivos que los negativos, pero tiene un fuerte margen de error, más en el primero que en el segundo. En promedio de todas las gráficas anteriores, su área bajo la curva es del 64 %, como lo comentando anteriormente, este modelo es el que tiene una métrica más baja comparada con las anteriores, y se debe a varios puntos uno de los primeros es que los conjuntos de datos no concuerden con el comportamiento que se requiere.

### Red Neuronal Artificial

Una red neuronal es una combinación de nodos de activación, los datos de entrada, recorren la red, se activan, desactivan, dejando pasar cierta cantidad de información para converger en una salida, la cual es un clasificador. La manera en cómo funciona, es que las señales de entrada, la información que recibe nuestra red neuronal artificial, son las variables independientes. Las entradas multiplicadas por los pesos, dando como resultado lineal datos de entrada y pesos, los pesos son aleatorios al inicio, pero las entradas están determinadas por el número de características que se definieron, es decir 22, esta función se determina:

$$X * W^t = (x_1, x_2, \dots, x_{22}) * (w_1, w_2, \dots, w_{22})^t = \sum_{i=1}^{22} x_i * w_i$$

Luego de ello se tiene la función de activación, este es un hiper-parámetro el cual, dependiendo del comportamiento de los datos, se determina de la siguiente forma

$$\phi\left(\sum_{i=1}^{22} x_i * w_i\right)$$

Esta función de activación es una función que transmite la información generada por las entradas, es decir son la manera de transmitir la información por las conexiones de salida. La información puede transmitirse sin modificaciones, lo que realizara es que los modelos sean no lineales, la función sigmoideal de activación es la mejor para este problema fue:

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

Desde ahí se parte del polinomio, que se requiere optimizar, lo cual hace que los pesos se actualicen cada vez que se itera, actualizándose, con la propagación hacia se compara si con el peso se corrió es igual a lo que debió predecir contra lo real en dado caso que no sea, de igual manera este, es un hyperparametro, el optimizador que se utilizó fue el descenso de gradiente.

Los demas hyper-parámetros son las épocas que es él número de iteraciones, los cuales actualizan tantas veces los pesos que adecuen a la salida, otra más es el tamaño de los bloques, esto se realiza para la actualización de pesos de la entrada de la red, cada cuando se van actualizando los pesos. Para el modelo se tomaron 100 épocas y cada 10 ejemplos la actualización de pesos, es decir, de cada diez características de cada iteración se actualizan todos los pesos.

Las redes neuronales tambien pueden tener capas ocultas donde los pesos pasan la información unos con mayor ponderación o menos, esto conforme a la actualización de los datos, este es un hyper-parámetro de igual forma para la red son dos capas ocultas la primera es 20 neuronas y la segunda de 16.

En la siguiente figura vemos, cómo queda nuestra red neuronal para ese ejemplo:

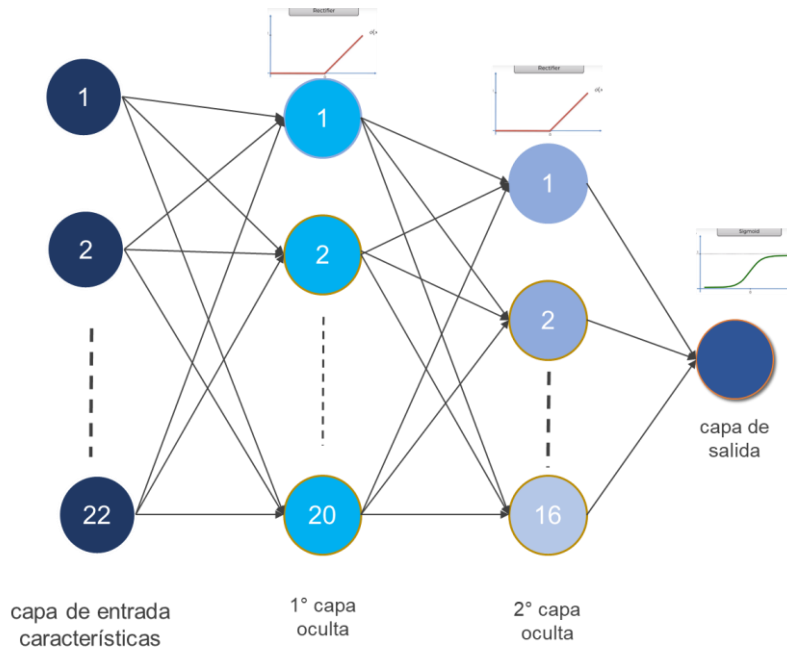


Figura 70 – Red Neuronal

La siguiente figura muestra el promedio de los datos, la matriz confusión:

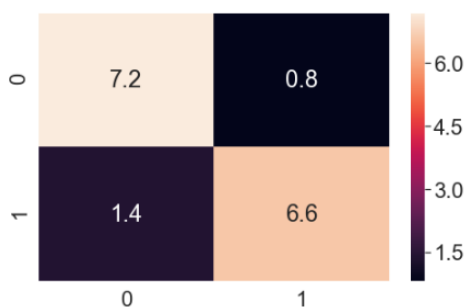


Figura 71 – Matriz de Confusión

Se puede ver, en la matriz de confusión que la mayoría de los datos de prueba, lo clasifica correctamente, el único defecto que se va inclinando es de falso positivo, en promedio casi dos pruebas que deberían de clasificarse como verdaderos se clasificaron como negativo.

Lo anterior también lo podemos ver en sus métricas las cuales son en recall es de 82 %, siendo esta la más baja en comparación de las otras, lo cual se debe a los falsos positivos:

Recall	0.825
Specificity	0.9
Accuracy	0.8625
Precision	0.8958
F1	0.8542

Figura 72 – Métricas ANN

Con las métricas pasadas obtenemos las curvas ROC, que se muestra en la siguiente figura:

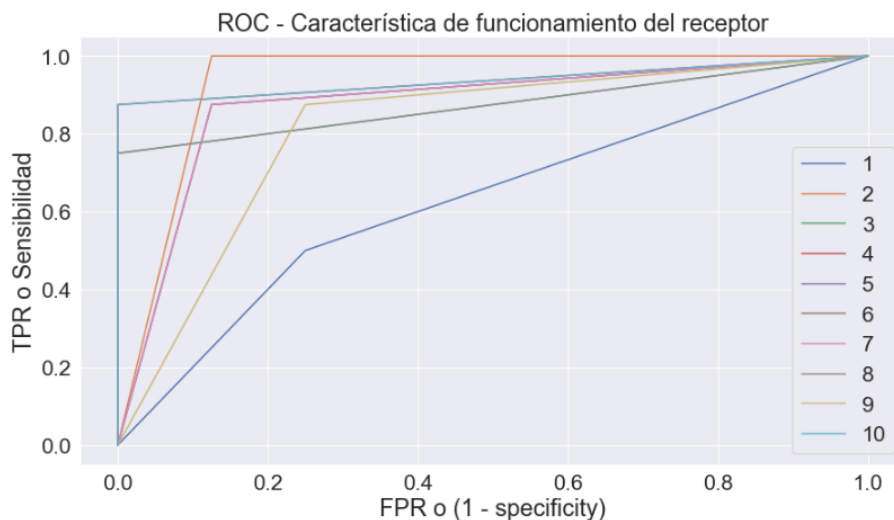


Figura 73 – Curvas del Modelo ROC

El promedio de los experimentos, es de **86 %**, esto es un rendimiento muy adecuado para un clasificador, y como lo podemos ver en la curva ROC, se ve como varios de los experimentos se clasificaron correctamente los ejemplos positivos como los ejemplos negativos.

## Máquinas de Soporte Vectorial Kernel Lineal

Este modelo, se verá con más de detalle en la siguiente sección dado que el kernel solo es el que cambia, las SVMs (*Support Vector Machines* por su traducción a las ingles) con kernel polinomial es el modelo que tuvo mejor rendimiento.

En esta apartado expondremos superficialmente y mostraremos los resultados de las SVM, es un conjunto de algoritmos de aprendizaje supervisado, funciona con un conjunto de datos de entrenamiento se etiquetan las clases, luego se entrena una SVM para construir un modelo que prediga la clase de la nueva muestra. Intuitivamente, una SVM, es un modelo que representa a los puntos de entrenamiento en el espacio, separando las 2 clases a espacios lo más amplio posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de 2 clases más cercanas, este el vector de soporte. Con ello se construye un o conjunto de hiperplano en un espacio de dimensiones muy altas, para su separación.

Los kernel son las funciones de cómo se construye un hiperplano, este un hyperparametro, la manera más simple es un plano recto o un hiperplano N-dimensional, aunque, este kernel no es muy bueno en la práctica por diferentes razones.

Los resultados de los diez experimentos, con el método de k-Fold en la matriz de confusión son:

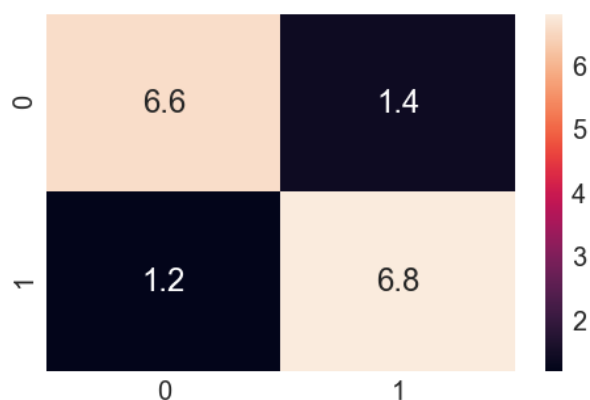


Figura 74 – Matriz de Confusión SVM

Como podemos ver, la matriz es un promedio correcto tanto para la clasificación de verdaderos positivos y falsos negativos, los errores que comete tanto para los falsos positivos como para los falsos negativos son casi simétricos, de igual manera podemos ver las métricas que se obtuvieron de la anterior matriz:

Recall	0.85
Specificity	0.825
Accuracy	0.8375
Precision	0.8444
F1	0.8418

Figura 75 – Métricas de SVM

Estas métricas nos indican que el clasificador, está separando bien las dos clases positivas y negativas, la métrica más baja es la especificidad donde es el error que cometió el modelo. En la siguiente figura vemos las curvas ROC de los ejemplos:

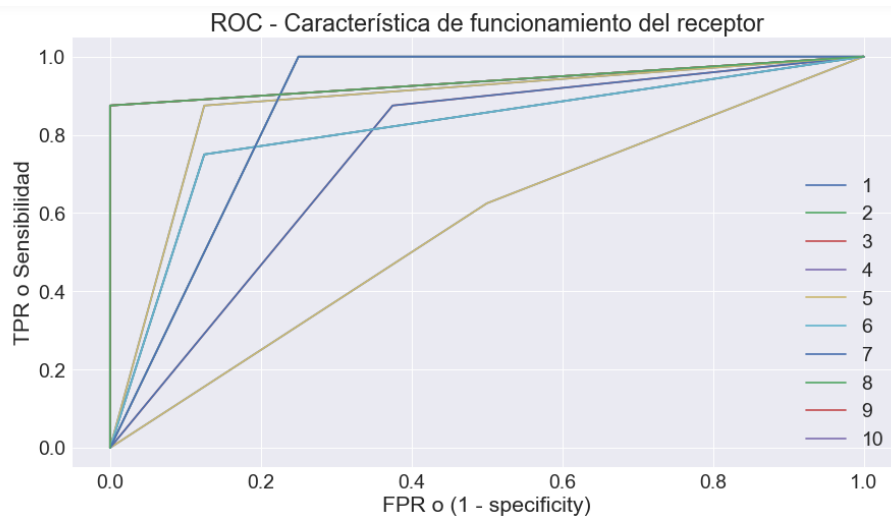


Figura 76 – Modelo ROC de SVM

El promedio de las AUC sobre las curvas ROC, es del **83 %** con lo cual podemos visualizar en la figura a continuación, además la mayoría de los experimentos quedaron aproximadamente del 80 %, el único experimento quedo del 62 % conforme a la sensibilidad es el caso especial para esos datos de prueba.

En resumen, los modelos que se ha descrito, se realiza una comparativa, con todas las métricas, la cual se puede observar en la siguiente imagen:

	Recall	Specificity	Accuracy	Precisión	F1	AUC
Regresión Logística	0.88	0.81	0.83	0.78	0.82	0.84
DGA	0.59	0.62	0.60	0.71	0.64	0.61
SVM lineal	0.85	0.82	0.83	0.84	0.84	0.83
SVM Polinomial	0.78	0.93	0.86	0.92	0.84	0.86
ANN	0.82	0.9	0.86	0.89	0.85	0.86

Figura 77 – Cuadro comparativo de los modelos experimentados

La anterior grafica nos muestras una comparación con todas las métricas, muestran el comportamiento detallado de los modelos, se puede ver algunos modelos tienen mejor

rendimiento en algunas métricas por ejemplo la regresión logística en la métrica de recall tiene mejor rendimiento. Pero para hacer una mejor elección se toma en cuenta la métrica de AUC, en esta parte la red neuronal artificial y el modelo de SVM polinomial, quedan con los mismos rendimientos, luego comenzamos hacer una comparación y el modelo de SVM, supero en decimales a los demas modelos, siendo este el modelo más adecuado para la clasificación de techos y no techos.

## SVM Polinomial

En este apartado veremos con un poco más detalle el algoritmo, desde su descripción matemática los hyperparametros y resultados que dieron en este modelo.

Como se mencionó anteriormente es un algoritmo que sirve para la clasificación y regresión, el objetivo es tratar de buscar el hiperplano que separa lo mejor posible las clases con el mayor margen que se pueda. Los SVMs funcionan principalmente para resolver problemas binarios, y según los datos para este modelo se clasifica, en margen duro, cuando tienen una separación linealmente separable, margen débil, cuando se permite un error de mezcla en las distintas separaciones.

La idea de los SVM es encontrar las variables  $a$ ,  $b$ ,  $c$  que separa las rectas con los vectores de soporte, como se ve en la siguiente figura:

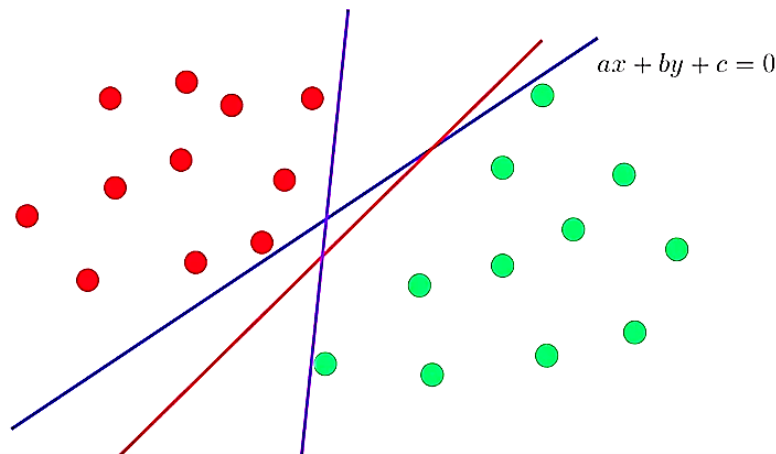


Figura 78 – Separación de clases de SVM Polinomial

La anterior figura es un ejemplo con dos características donde están dos clases, es complejo poder hacer una gráfica con las 22 variables de las características dado que se vuelve irrelevante, lo se debe de encontrar es el máximo margen de los vectores de soporte, que son los límites entre cada una, como se puede ver en el margen máximo:

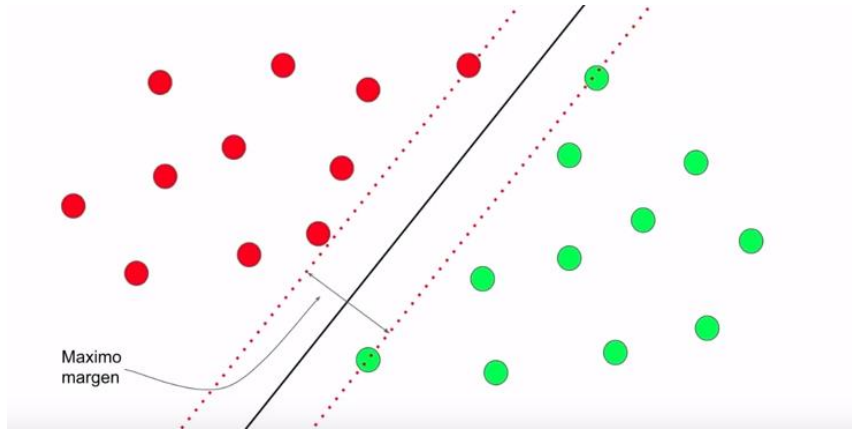


Figura 79 – Vectores de soporte ejemplo

Para encontrar los vectores de soporte, el problema es optimizar, en las entradas, tenemos las primeras clases de 1 a  $X_i$ , la segunda clase sería,  $-1$  a  $X_i$ , donde  $i$  serían todas mis observaciones, es decir mis características, los pesos que serían el vector de dirección, con esto tendríamos:

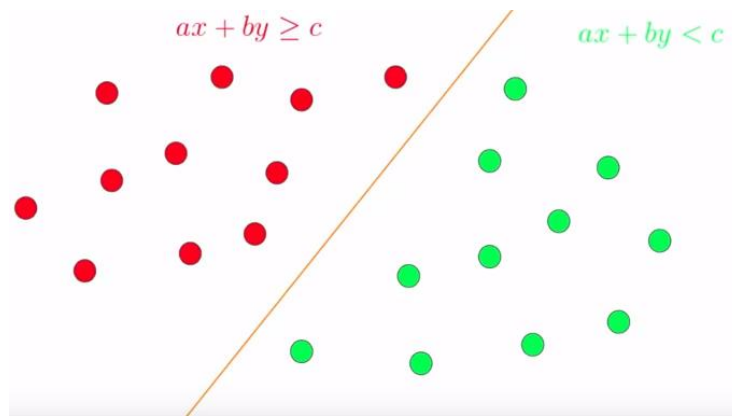


Figura 80 – Ejemplificación de separación de clases

Si creamos dos líneas imaginarias para poder tener los ejes vectores, los vectores serían la ecuación de las líneas donde 1 y -1 son las clases, como se muestra en la figura siguiente:

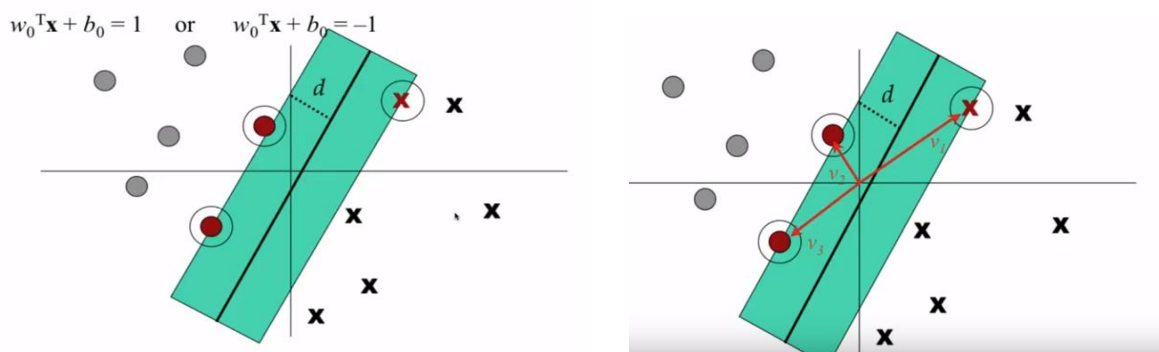


Figura 81 – Vectores de distancia

Los anchos de las clases serian dos veces  $d$ , el objetivo sería máxima ese ancho entre esos tres vectores de soporte, de esta manera definimos los vectores de  $d$ , lo agregamos para igualar el problema, así caracterizamos ejemplo:

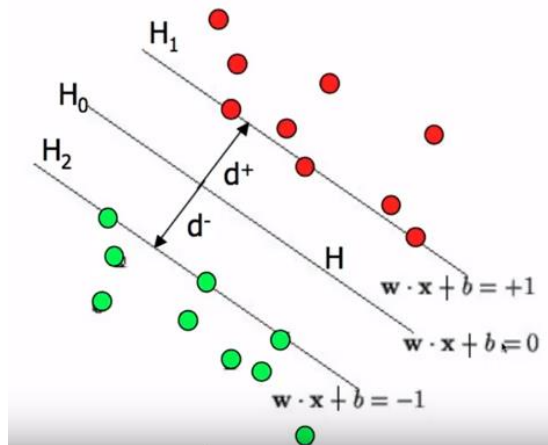


Figura 82 – Descripción matemática de vectores de soporte

Entonces si podemos obtener la distancia de un punto hacia el vector de dirección ( $H_0$ ), la fórmula de la línea  $l$  y el punto  $P$ :

$$L: ax + by + c = 0$$

$$P(X_0, Y_0)$$

$$d(LP) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

Para sacar la clase 1 de la ecuación quedarías así:  $\frac{|w \cdot x + b|}{|w|} = \frac{1}{|w|}$

Para máximas el ancho de  $d$  que se tiene que realizar en la distancia es minimizar  $w$ , entonces el problema de optimización quedaría, luego se puede poner esas dos condiciones, quedando:

$$\min |w|$$

Sujeto a:

$$\left. \begin{array}{l} \bar{x}_i \bar{w} + b \geq +1 : y = +1 \\ \bar{x}_i \bar{w} + b \leq -1 : y = -1 \end{array} \right\} y_i(\bar{x}_i \bar{w}) \geq -1$$

Nuestra funcion quedaría:

$$\sqrt{w_1^2 + w_2^2 + \dots + w_{22}^2}$$

Lo que se derivaría será dentro para trabajar directamente con  $w_i$ :

$$\min \frac{1}{2} |w|^2$$

Sujeto a



$$y_i(w * x_i + b) - 1 = 0$$

Para poder resolver el problema generamos la reducción, primero utilizamos la función escalar de Lagrangiano, se utiliza para encontrar los máximos y mínimos de funciones de múltiples restricciones, este método reduce el problema restringido con  $n$  variables a uno sin restricciones de  $n + k$  variables, donde  $k$  es igual al número de restricciones, en cuyas ecuaciones pueden ser resueltas más fácilmente :

$$L(\vec{w}_1, b) = \frac{1}{2} |w|^2 - \sum_{i=1}^N \lambda_i [y_i(w * x_i + b) - 1]$$

Luego de ello, se escribe de otra manera donde los separamos en tres sumatorias, y podemos encontrar la solución, en el caso la minimización sería,  $L$  con respecto a  $w$  y con respecto a  $b$ , por la forma en cómo funciona el problema primal, se resuelve haciendo los pesos  $\vec{w}$  con respecto a lambda, a los parámetros  $x$  y la predicciones que serían las  $y$ .

Luego de encontrar el primal, obtenemos el dual, sacando con respecto a lambda, de esta manera podemos encontrar el hiperplano, con la parte de la función de costo:

$$L(\lambda_i) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \lambda_i \lambda_j y_i y_j (x_i * x_j)$$

Ahora, en el caso que es kernel polinomial solo cambia desde el punto de maximización, desde la parte dual, el kernel que utilizamos:

$$K(x, z) = \langle x, z \rangle^d$$

Donde  $d$  es grado del polinomio que se debe de tener es el polinomio que se tiene, es de grado 1 este un hiperparámetro, se probaron diferentes de los cuales el polinomio de grado 1 es el que dio los resultados, a continuación, presentamos la parte dual, de cómo queda la ecuación, con el kernel polinomial:

$$\text{Maximizar } L(\alpha) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i x_j)^d + \sum_i \alpha_i$$

$$b = y_i - w^T \varphi(X_i) = y_i - \sum_j y_j \alpha_j K(x_i x_j)^d$$

$$f(x) = \text{sgn} \left( \sum_i y_i \alpha_i K(x_i x_j)^d + b \right)$$

El alpha sustituye a lambda dado que otros autores también lo utilizan (Huang, 2018). En la anterior ecuación vemos como incluye el kernel tanto los vectores de soporte como, el hiperplano transformándolo,  $\text{sgn}$  son abreviaturas para una función sigmoideal, es decir que en esta parte conforme a la salida se pone 0 o 1 donde ya clasifica para positivo o negativo, dado que la función sigmoideal. Ahora la parte primal de nuestra ecuación, tenemos varios hiperparámetros que van buscando el margen, por ejemplo, la variable  $\epsilon$  va estar

cambiando dado que hay varios datos que están cruzando la frontera, de igual manera otro hyperparametro es la funcion de en l parte primal:

$$\text{Minimizar } L(W) = \frac{1}{2} w^T w + C \sum \varepsilon_i$$

$$\text{Sujeto a: } y_i(w^T \phi(x_i) + b) \geq 1 - \varepsilon_i$$

Con ello, definimos los dos con, el kernel polinomial, una de las ventajas que tiene SVM es que no tiene una actualización propia de parámetros que cambié o varíe mucho entre pruebas sus resultados, es decir si viene un dato nuevo no cambia sus métricas que dio por que lo tambien con todo el conjunto de datos de entrenamiento. En la siguiente figura tenemos los resultados de los 10 experimentos cada una tiene 20 datos de test, de ello se sacó un promedio para dar un resultado más adecuado:

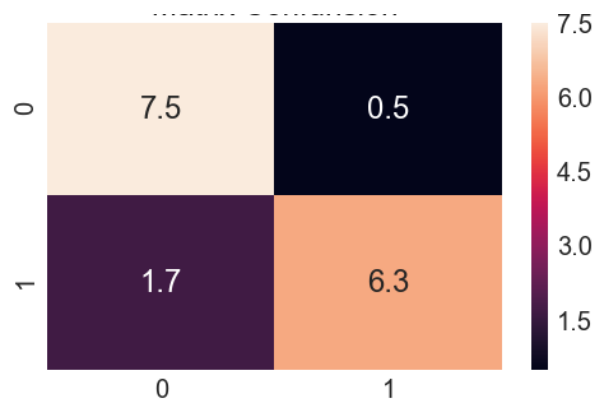


Figura 83 – Matriz de confusión SVM Polinomial

Como podemos ver en la matriz de confusión, tanto los ejemplos positivos como los ejemplos negativos, tiene en general una correcta clasificación, donde comenta más errores casi dos pruebas es la parte de falsos positivos, esto tambien los podemos ver en el siguiente cuadro:

Recall	0.7875
Specificity	0.9375
Accuracy	0.8625
Precision	0.9263
F1	0.8461

Figura 84 – Métricas SVM Polinomial

Como lo mencionado, la parte de recall es la métrica más baja por los errores de los falsos positivos, en las otras métricas, se tiene un rendimiento elevado, la especificidad tiene un aumento ligeramente, demostrando como lo clasificado, separa más la clase falsa. En la siguiente figura veremos las curvas ROC de los experimentos:

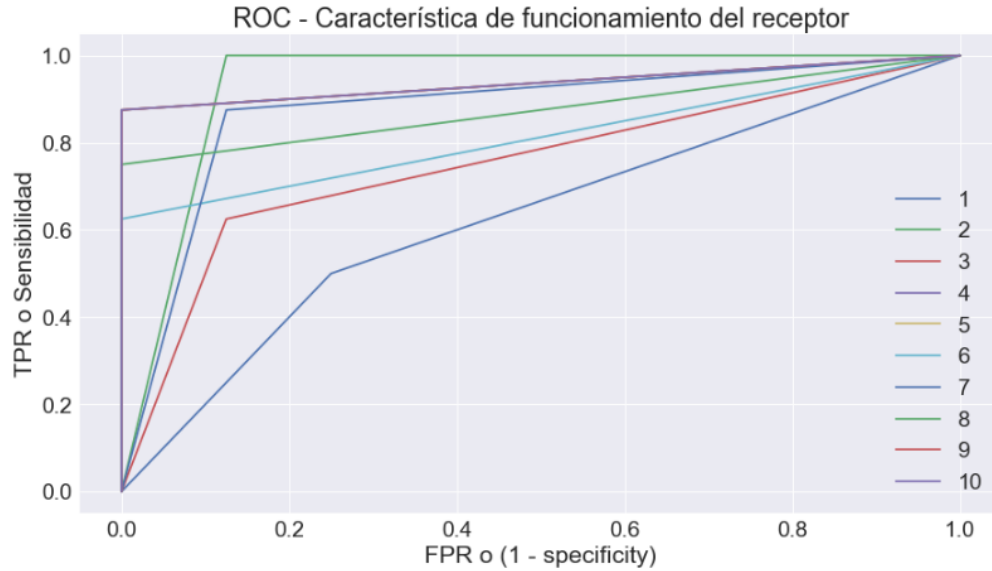


Figura 85 – Experimentos Curvas ROC SVM polinomial

El promedio de las curvas ROC es de **86 %**, es un rendimiento adecuado para la aplicación de los nuevos datos, podemos apreciar que en la mayoría de los experimentos los AUC, estuvieron cerca, pero el séptimo experimento fue el que estuvo más bajo.

En conclusión, de este apartado, el modelo de SVM con kernel polinomial por las métricas y encajo de mejor manera para predicción de techos factibles y los que nos son.

# Capítulo IV

## Aplicación

En este capítulo podremos ver el diagrama del algoritmo integrado y como funciona, la aplicación de este a diferentes muestras de las secciones de la ciudad, por último, se analizará la prueba experimental en una instalación para la comparativa del modelo la predicción contra lo real.

### Algoritmo integrado

Luego de distinguir tanto los métodos de segmentación como el algoritmo desarrollado el cual elimina todo lo que no sea techo, este integra al algoritmo clasificador, el cual arroja el mapa a nuestro clasificador ya entrenado previamente, esto se puede apreciar en el siguiente diagrama de flujo:

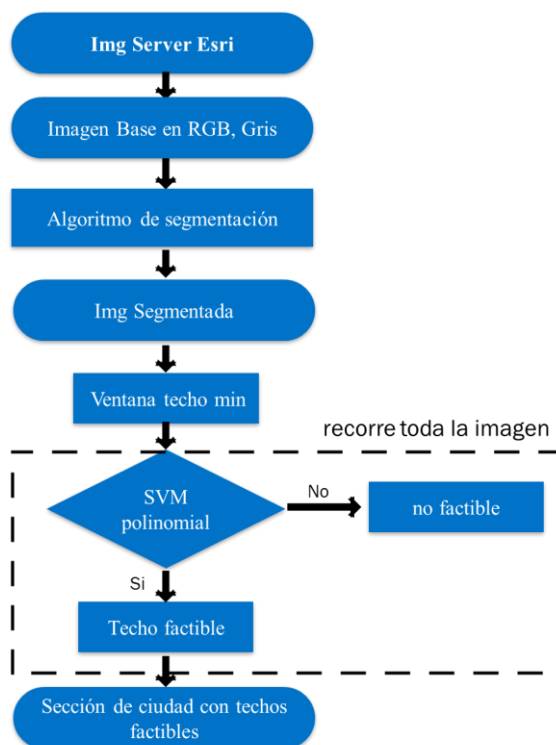


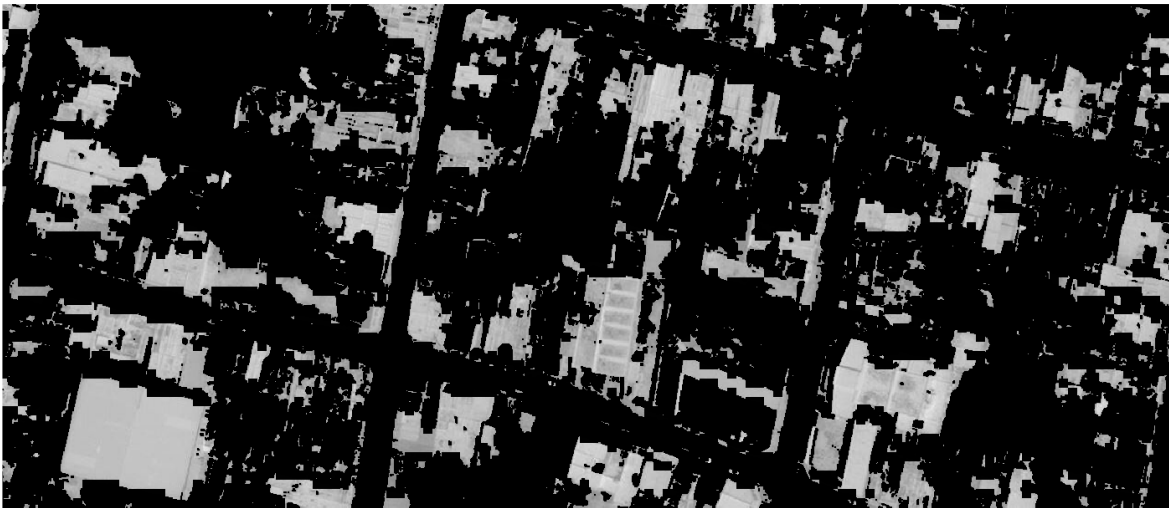
Figura 86 - Diagrama de flujo Algoritmo Integrado

El anterior diagrama demuestra el proceso completo para seleccionar si un techo es factible para la instalación de paneles fotovoltaicos o no, el proceso inicio desde la obtención de la

imagen de los servidores de datos libres de la plataforma Esri, se obtiene una sección de la ciudad, luego se procesa por el algoritmo de segmentación, el desarrollo de este se encuentra en el capítulo 2, la imagen que nos arroja el algoritmo, es una imagen de una sección de la ciudad con todo lo que no sea techo, es decir, vegetación, calles, obstáculos y demás. Se parte de la imagen base segmentada en esta parte, el modelo de clasificación recorre toda la sección de la ciudad, con una ventana de 5 x 5 píxeles, el modelo ya entrenado de SVM con kernel polinomial, clasifica las secciones que nos factibles para la instalación o no, con un enfoque de textura de la ventana a clasifica. Por último cuando el modelo ya termina tenemos un mapa base para que los puntos con colores de grises son los espacios que sirven para la instalación de paneles fotovoltaicos.

## Aplicación

En el capítulo dos, se analizó la aplicación del algoritmo de segmentación en diferentes secciones de ciudades, en este apartado también veremos las aplicaciones de las mismas, pero en este caso con la aplicación del algoritmo integrado, se podrá analizar las muestras de ciudad.



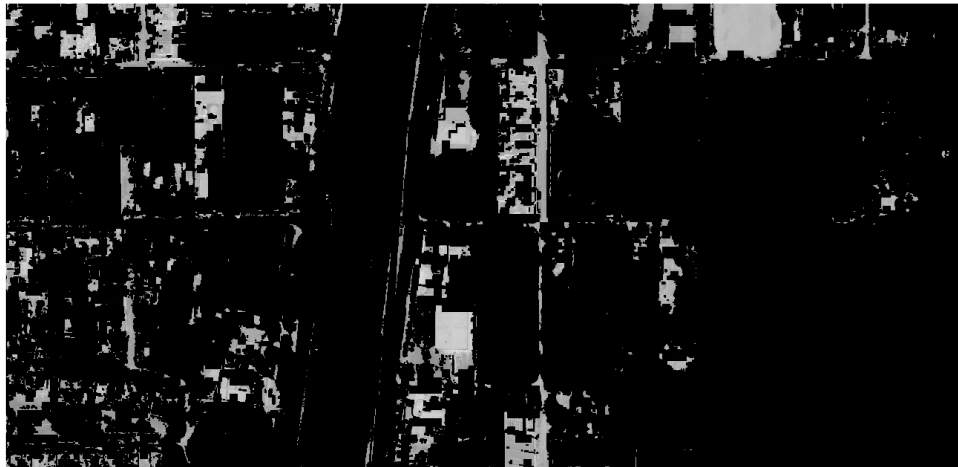
*Figura 87 – Muestra 1 Aplicación del modelo algoritmo integrado*

Para hacer una comparación correcta primero revisar la figura 16 donde se encuentra en el capítulo 2, en esta aplicación la parte del techo se tienen más elementos eliminando, más partes no factibles, dejando solo techos con texturas homogéneas, cabe decir que de esta imagen no tiene un post-tratamiento para hacer cuadrados las áreas factibles, de igual manera los píxeles que está en diferentes partes eliminarlos porque solo genera ruido en la imagen.



*Figura 88 – Muestra 2 Aplicación del modelo algoritmo integrado*

En la anterior muestra podemos observar cambios considerables de los puntos grises que en este caso son los techos factibles, en comparación con la figura 34, y la figura 35, el algoritmo elimino todo lo que tenía obstáculos como se puede percibir, un punto importante para notar es la estructura del edificio se puede ver que tiene niveles y el algoritmo integrado no tomo como factible la parte que genera sombra de la edificación este un punto a favor dado que la sombra en un panel fotovoltaico no permite la generación.



*Figura 89 – Muestra 3 Aplicación del modelo algoritmo integrado*

La anterior muestra es el límite de la ciudad separando por una el anillo periférico, para ver la imagen original ver la figura 36, la imagen de la derecha la sección de la ciudad era más campo con pocas casas, el algoritmo de segmentación había dejado ciertas secciones, pero el modelo de clasificación, elimino por completo dejando solo los puntos factibles en toda la sección de la ciudad, tambien otro punto era ver como se comportaba con carreteras anchas,

en esta muestra vemos que no toma como factible para instalación lo cual demuestra su funcionamiento para esos casos.



*Figura 90 – Muestra 4 Aplicación del modelo algoritmo integrado*

En la anterior muestra, se tiene una sección de la ciudad donde era una glorieta con varias casas por la derecha una avenida con una edificación grande, para ver la muestra original ver la figura 38, en esta sección de la ciudad se aplicó en la glorieta porque al ojo humano se tenía una la visualización de una textura homogénea la cual el algoritmo de segmentación no lo elimino pensando que era techo, pero el modelo de clasificación si lo pudo eliminar, dejando ciertas secciones, que al momento de ver los mínimo que debe tener una casa para la instalación se elimina.



*Figura 91 – Muestra 5 Aplicación del modelo algoritmo integrado*

En la anterior muestra es la sección de la ciudad de un aeropuerto en más detalle son dos puertas de abordaje como sus terminales mostrando dos aviones, revisar la figura 40 imagen original, este es un caso especial y donde no interpreta bien el algoritmo integrado de los datos de entrada, el lado donde se encuentra la instalación de aeropuerto para la espera de pasajeros clasifica correctamente los techos dejando las áreas factibles, pero la pista del aeropuerto es donde no clasifica correctamente comenzando lo toma como techo factible,

aunque es un término para cuestionarse dado que la textura y es sabido correctamente que la pista de los aeropuertos no tienen que tener un obstáculo, con esto podemos llegar a la conclusión donde se pueden instalar correctamente, y que es un área factible para instalar.

## Prueba Experimental

Para probar el modelo en un techo real, se realizó una prueba experimental en un techo donde se pudiera medir la dimensión del techo, el objetivo es ver cuando es lo que predice el modelo y cuál es el área que el modelo arroja que pudiera ser factible para la instalación de paneles fotovoltaicos.

El experimento se realizó en la Facultad de Ingeniería de la Universidad Autónoma de Yucatán, Mérida, Yucatán, México su ubicación exacta con altitud y latitud es de 21.048577, -89.639942. Los pasos del experimento fueron:

- Obtención de imagen con mayor calidad.
- Levantamiento de medición de área del edificio a comparar.
- Cálculo de los píxeles a metros

La obtención del mapa con mayor calidad, el proceso en conjunto con otro grupo de investigación fue realizado con un dron el cual mapeo todas las instalaciones, la resolución obtenida fue de 0.3 m por píxel, el proceso fue la programación del dron por sectores para que luego se haga la miscelánea de dicha imagen:

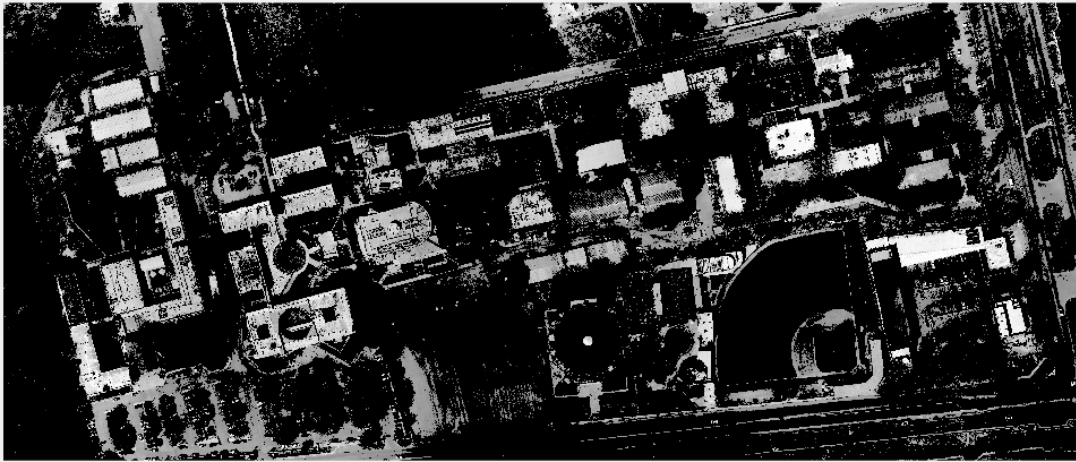


*Figura 92 – Miscelánea Facultad de ingeniería*

Luego de ello, se recorta la imagen y lo pasamos aplicamos el algoritmo integrado, con esto tendremos los techos factibles para las instalaciones, la siguiente figura muestra como los resultados de la aplicación:



Podemos observar que el algoritmo no está clasificando correctamente los techos de color rojo por la una resolución, también la parte del estacionamiento está como factible, mayor los demás techos los clasifica correctamente.



*Figura 93 – Facultad aplicado con el algoritmo integrado*

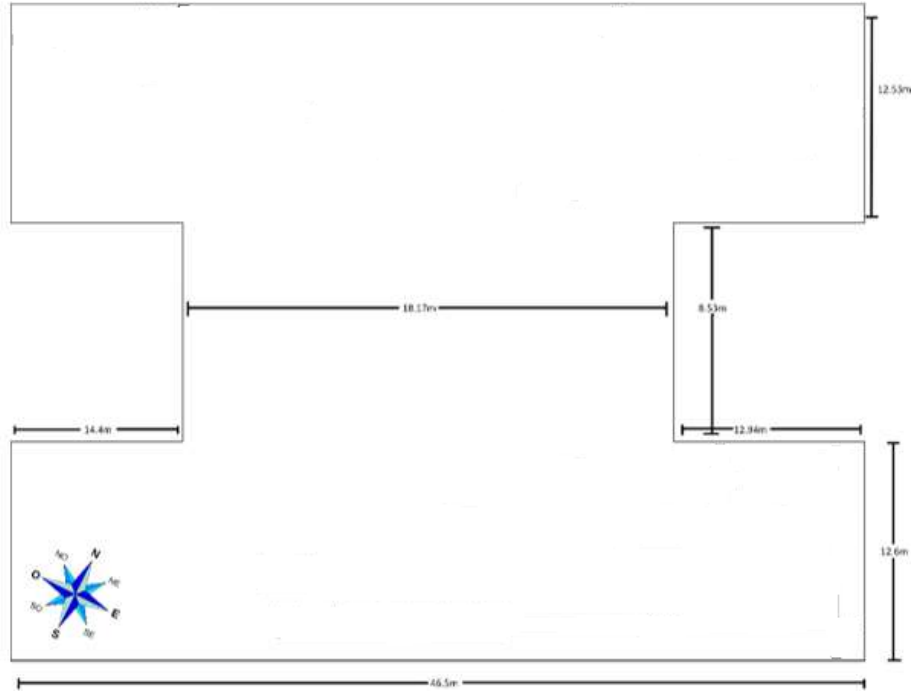
El segundo paso, el levantamiento se realizó de manera manual, es decir con instrumentos de medición para metro, se midió de borde a borde todo el techo seleccionado. la siguiente figura muestra el levantamiento de las medidas:



*Figura 94 – Levantamiento de medidas*

se escogió este edificio por su forma irregular, no el típico techo rectángulo, además de tener varias características, como la instalación de paneles fotovoltaicos los cuales son obstáculos para la instalación de nuevos paneles fotovoltaicos, no tiene ninguna sombra proyectada como ningún obstáculo visible.

Luego de hacer el levantamiento de las medidas se realizó el croquis como se ve en la siguiente figura:



*Figura 95 – Croquis del Edificio a evaluar*

El anterior croquis las medidas que obtenemos de toda el área cuadrada es de 1255.31 metros cuadrados, estas son las medidas de borde a borde, cabe recalcar que quitaron 2 metros cuadrados por algunos obstáculos que vieron en el techo donde no se podría instalar paneles. La siguiente figura es la imagen original del edificio



*Figura 95 – Imagen original de edificio a evaluar*

Esta imagen base se procesa el algoritmo integrado para poder segmentar todo lo que no sea techo, por lo tanto, la siguiente figura muestra del lado izquierdo la imagen segmentada y del lado derecho y la imagen clasificada:

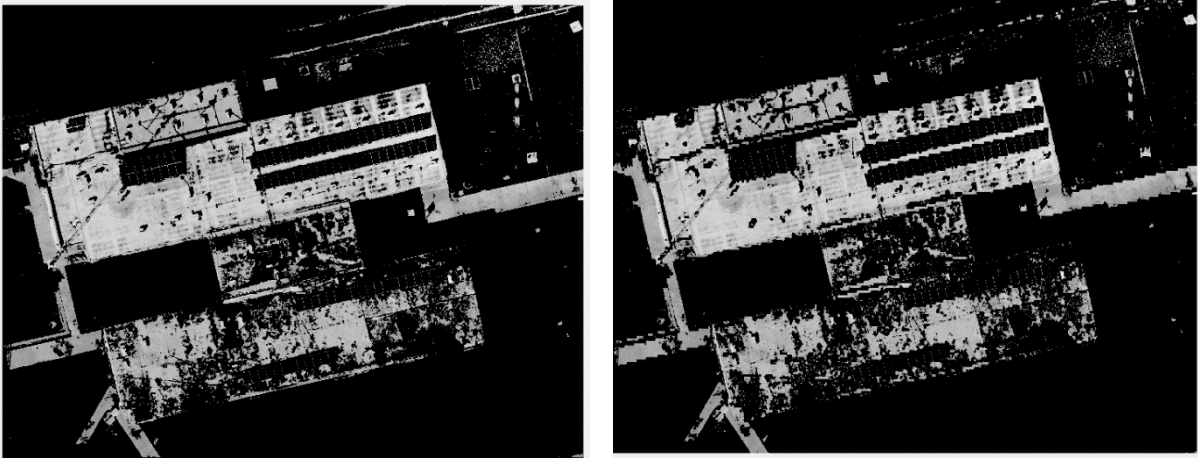


Figura 96 – a) Imagen Segmentada, b) Imagen Clasificada

fotovoltaicos ya instalados como área no factible para su instalación, de igual manera los demás obstáculos, como es una imagen de alta resolución el cambio de textura en el edificio hace que la lo tome como si fuera un obstáculo, la percepción con el ojo humano también pone en duda si es un techo factible para la instalación o no.

Haciendo el cálculo de los píxeles la imagen nos da que el área factible para instalar en metros cuadrados es de 778.94 mientras que el área en el levantamiento es de 1255.31 metros cuadrados, por lo tanto, la precisión que se tiene en porcentaje es de 64 %.

#### Trabajos futuros

- Imágenes multiespectrales, ver el relieve de diferentes materiales, hacer clasificación
- Como generar mapas
- Ver la resolución
- Mapas de sombra

# Capítulo V

## Conclusiones

El ultimo capitulo veremos diferentes puntos importantes donde converge todo el proyecto de investigacion, las conclusiones del proyecto, los objetivos logrados y los trabajos a futuro.

### Conclusiones

- El algoritmo integrado, cumple con la clasificación de techos para las áreas factibles de instalación, pero con un enfoque más a clasificar las áreas factibles, mientras el techo no tenga obstáculos o no tenga un cambio de color hacia la parte más oscura, el algoritmo cumple decidir si la imagen digital es un techo o no.
- La combinación de tratamiento de imágenes digitales, con SVM, puede competir con las redes neuronales profundas, la ventaja de no utilizar el base de datos puede ser de una cantidad reducida.
- En proyecto de investigacion recae su eje principal en la textura, es decir casi toda la solución planteada en los atributos de textura de la imagen, pero como se pudo ver en las aplicaciones de las muestras, se requiere tambien una información para contextualizar la clasificación, por ejemplo, información adicional de la misma ciudad como lugares no instalables, parques, aeropuertos, estadios, etc. De esta manera se ahorra tiempo de procesamiento y se enfoca más en problemas de techos.
- El algoritmo de segmentación, descarta correctamente casi todo lo que no sea techo, pero, como lo explicado el algoritmo toma toda la imagen y de esta manera segmenta varios puntos de la imagen descartándolo, para una mejor segmentación se debería de aplicar el algoritmo de segmentación, pero esto con ella ciertos requisitos la obtención de la imagen de la toda la ciudad en alta resolución y un hardware de alto procesamiento.
- En el apartado de experimentación de diferentes modelos, se pudo visualizar que la red neuronal tambien podría ser una buena opción para la clasificación, con nuevos datos y un crecimiento considerable del conjunto de datos se puede recorrer todos los modelos para ver cuál de los modelos ahora genera más rendimiento para la clasificación, esa es unas grandes ventajas que la metodología que se usó para la construcción automatizada del modelo.
- En el capítulo cuatro la sección ver lo real contra lo que predijo el modelo, se tuvo el 64 %, este número no determina la validación del modelo dado que para determinarlo

se tendría que tener más muestras aleatorias, lo que se pudo observar del contrario que se intuía es al tener una imagen con resolución de 0.3 metros por pixel, se vuelve más complicado para el algoritmo de clasificación dado que la textura de los techos se vuelven más heterogámicas y el algoritmo no percibe si es un obstáculo para la instalación o un macha en el techo, por lo tanto es recomendable trabajar con intensidad de 0.5 metros por pixel.

## Objetivos logrados

- Se logró el objetivo principal, el desarrollo de un modelo que muestre el potencial de recurso solar, basando en imágenes satelitales, mostrando las áreas factibles para la instalación.
- Se creó con éxito el algoritmo de segmentación cuya función descarta todo lo que no sea techo.
- Se desarrolló un modelo de IA con éxito desde la base de datos que es *open source*, las características esenciales y sus métricas de evaluación.
- Se pudo integrar los algoritmos de segmentación y clasificación para tener un modelo completo.
- Se realizó por completo, las pruebas de experimentación para la presentación de los modelos.

## Trabajos a Futuro

- Para investigaciones posteriores, es poder obtener imágenes multiespectrales de la ciudad, para poder mejorar el algoritmo de segmentación dado que en el estado del arte se puede decantar por el reflejo de los materiales de la luz hacia el satélite, con ello podemos eliminar las calles y otras partes de la ciudad que se parecen a un techo.
- El SIG, fue uno de los objetivos que se deja para la posterior investigación, dado que la resolución de la ciudad con las imágenes sectorizadas como calles, y demás partes de la ciudad están desfasadas por la resolución, además que para algunas ciudades por sus gobiernos tiene capas de mapas vectorizados de predios GIS, esto para que solo se quede los predios registrados, y sea más la clasificación.
- La simulación de sombras, para ver las áreas factibles, es un punto importante para posteriores investigaciones, con el fin de validar con más detalle el área factible y con el mejor rendimiento de energía.
- Calcular la producción de una ciudad, se dejaría para la siguiente investigación, dado que se requiere la imagen de la ciudad en alta resolución, la simulación de sombras y un estudio de radicación en la ciudad a calcular.

# Referencias

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al. (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available online: tensorflow.org
- Bishop, C. (2006). Pattern recognition and machine learning (Vol. 3). California, EU: Microsoft Research Ltd.
- Bouckaert, R. (2003) Choosing between two learning algorithms based on calibrated tests. In Proceedings of 20th International Conference on Machine Learning. pp. 51–58.
- C. Shannon. (1948) “A mathematical theory of communication” Bell System Technical Journal, vol. 27, pp. 379-423.
- CEPAL (Comisión Económica Para América Latina y el Caribe). (2018). Informe nacional de monitoreo de la eficiencia energética del México: Naciones Unidas, 18, 200. Consultado enero 2020, De Repositorio Cepal.org Base de datos.
- Chang, C., Wang, H., Wang, Y., (2019). Advanced Remote Sensing Technology for Coastal Environment, Disasters, and Infrastructure. IntechOpen. Recuperado de www.intechopen.com
- Chio, C., & Freeman, D. (2018). Machine Learning & Security (Vol. 1). California, EU: Octal Publishing, Inc.
- Christianini, N., & Shawe-Taylor, C. (2000). An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University Press.
- Clemente, J., Gay García, A., Quintana Solórzano, F. (2016). 21 Visiones de la cop21. El acuerdo de París: retos y áreas de oportunidad para su implementar, 330.
- Datta, R., et al. (2008). Image Retrieval: Ideas, Influences, and Trends of the New Age. ACM Computing Surveys, Vol. 40, No. 2.
- Di Pietro, S. (2019). Acuerdo de París: ¿Nuevos compromisos con el medio ambiente o nuevas oportunidades de negocio? Estudios Internacionales, 51(192), 57-70. doi:10.5354/0719-3769.2019.52814
- Domingos, P. (2015). The Master Algorithm: How machine learning is reshaping how we live. pp. 246–7.
- Evangelopoulos, A., Maragos G. (2005). Soil Image Segmentation and Texture Analysis: A Computer Vision Approach. IEEE Geoscience and Remote Sensing Letters, 5, (4).
- Fan, R.-E., P.-H. Chen, and C.-J. Lin. (2005). “Working set selection using second order information for training support vector machines.” Journal of Machine Learning Research, Vol. 6.
- Gomez Allende, D. (2017). Reconocimiento de formas y visión artificial (3.a ed., Vol. 1). Madrid, España: Editorial RA-MA.
- Gonzaga Aguilar, C. (2014). Aplicación de índices de vegetación derivados de imágenes satelitales Landsat 7 ETM+ y ASTER para la caracterización de la

cobertura vegetal en la zona centro de la provincia de Loja, Ecuador. Consultado enero del 2020.

- Gonzaga Aguilar, C. (2014). Aplicación de índices de vegetación derivados de imágenes satelitales Landsat 7 ETM+ y ASTER para la caracterización de la cobertura vegetal en la zona centro de la provincia de Loja, México. Consultado en enero del 2020.
- Gonzalez, R., Woods E., & Eddins S. (2009)., Digital Image Processing Using MATLAB, Gatesmark Publishing.
- Gonzalez, R., Woods, R. (1992). Digital Image Processing. Addison-Wesley, Reading Mass 1992.
- Google Inc. (2015). Project Sunroof [Software Web Application]. Recuperado de <https://www.google.com/get/sunroof>
- Hall, P., & Gill, N. (2018). An Introduction to Machine Learning Intepretability (Oreilly ed., Vol. 1). California, EU: Octal Publishing Inc.
- Hastie, T., Tibshirani, R., Friedman, J. (2008). The Elements of Statistical Learning, Second Edition. NY: Springer.
- Hawkins, J. & Blakeslee, S. 2016. On Intelligence. s.l.: Times Books.
- Huang, Z. (2018). "The two books helping China's Xi Jinping understand artificial intelligence". Quartz.
- Kecman V., Huang, T., and Vogt, M. (2005). "Iterative Single Data Algorithm for Training Kernel Machines from Huge Data Sets: Theory and Performance." Support Vector Machines: Theory and Applications. Edited by Lipo Wang, 255–274. Berlin: Springer-Verlag.
- La Seran Palomino, N., Contreras, W., & Elena Ruiz, M. (2010). Procesamiento Digital de textura: Técnicas utilizadas en aplicaciones actuales de CBIR. Investigación de Sistemas e informática, 7.
- Li, G., (2010), Improved watershed segmentation with optimal scale based on ordered dither halftone and mutual informational, Computer Science and Information Technology, IEEE International Conference.
- M. Sezgin & B. Sankur. (2004). "Survey over image thresholding techniques and quantitative performance evaluation". Journal of Electronic Imaging. 13 (1).
- Mapwell, MIT, & Consultora MoDE. (2013). Solarsystem (beta) [Software Web Application]. Recuperado de <https://www.mapdwell.com/>
- Maurer, C., Rensheng Q., & Vijay R. (2003). "A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 2.
- MoDe Studio et MIT. (2013). Mapdwell. <https://www.mapdwell.com>. Febrero 2020, De Mabwell Solar System Base de datos.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundation of Machine Learning (Vol. 1). Massachusetts, EU: Adaptive Computation and Machine Learning.

- Nielsen, M. (2019). *Neural Networks and Deep Learning* (Cognitive Sciences ed., Vol. 3). Recuperado de <http://neuralnetworksanddeeplearning.com/>
- Otsu, N. (1979). "A Threshold Selection Method from Gray-Level Histograms." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 9, No. 1.
- Paglieroni, D., (1992). "Distance Transforms: Properties and Machine Vision Applications," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, Vol. 54, No. 1.
- Patterson, J., & Gibson, A. (2017). *Deep learning a practitioner's approach* (vol. 1). California, EU: o'reilly media, inc.
- Poynton, C. (2003). *Digital Video and HDTV: Algorithms and Interfaces*. Morgan Kaufmann.
- Sánchez, F., Vélez, A., Moreno, B. y Esteban, L. (2010). "Introducing Algorithm Design Techniques in Undergraduate Digital Image Processing Courses" *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 15, n°. 5.
- Scholkopf, B., Platt, J. C., Shawe-Taylor, A. J. & Williamson, C. (2001). "Estimating the Support of a High-Dimensional Distribution." *Neural Comput.*, Vol. 13, Number 7.
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms* (Vol. 1). Cambridge, Inglaterra: Cambridge University Press.
- Subsecretaría de Electricidad. (2014). "Estadísticas del sector eléctrico e Indicadores de CFE y LFC", Sector Eléctrico Nacional, <http://egob2.energia.gob.mx/portal/electricidad.html>
- Thorat, S., Rajendra, Y., & Karbhari, K., & Mehrotra, S. (2015). Estimation of Crop and Forest Areas using Expert System based Knowledge Classifier Approach for Aurangabad District. *International Journal of Computer Applications*. 121.
- Velez Serrando, J., Moreno Díaz, A., Sanchez Calle, A., & Sanchez-Martín, J. (2003). *Visión Artificial y Procesamiento Digital de una Imagen* (IEEE Books ed., Vol. 3). Ciudad de México, México: IEEE Books.
- Vitorino, G., Caamal, A., Monjaras, O. (2016). *SIG para análisis socio-energético hacia un panorama renovable en Cancún*. Universidad del Caribe: Departamento Ciencias Básicas e Ingeniera.
- Wong, S., Zhu, M., Liu, R., Lu, R. & Peng, J. (2016). Estimation of Hong Kong's solar energy potential using GIS and remote sensing technologies. *Renewable Energy*, 99, 325–335.
- Yeo, I., & Johnson, I. (2000). "A new family of power transformations to improve normality or symmetry." *Biometrika*, 87(4).
- Zou, K., O'Malley A., Mauri L. (2007). Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation*, 6;115(5):654-7.