

UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS
INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS
CENTRO DE INVESTIGACIÓN EN CIENCIAS

”SEGMENTACIÓN AUTOMÁTICA DE AISLADORES
ELÉCTRICOS USANDO POCAS IMÁGENES PARA EL
ENTRENAMIENTO”

TESIS

QUE PARA OBTENER EL GRADO DE
LICENCIADO EN CIENCIAS

PRESENTA

FRANCISCO AQUINO FLORES

DIRECTOR DE TESIS:

Dr. Juan Manuel Rendón Mancha

SINODALES:

Dr. Jorge Hermosillo Valadez (Presidente)

Dra. Lorena Díaz González (Secretario)

Dr. Jorge Alberto Fuentes Pacheco (Suplente)

Dr. Audberto Reyes Rosas (Suplente)

Cuernavaca, Morelos

Marzo 2022

Índice general

1. Introducción	3
1.1. Descripción del Problema	3
1.2. Justificación	3
1.3. Hipótesis	4
1.4. Objetivo general	4
1.5. Objetivos específicos	4
1.6. Alcances	4
1.7. Limitaciones	5
2. Marco Teórico	6
2.1. ¿Qué es una red neuronal artificial?	6
2.2. ¿Qué es una red neuronal convolucional?	7
2.3. Funciones de activación	8
2.4. Segmentación de imágenes	11
2.5. Definición desde el punto de vista intuitivo	12
2.6. Acomodo espacial	12
2.7. Arquitectura <i>encoder-decoder</i>	13
2.8. Capa de <i>pooling</i>	13
2.9. <i>Pooling</i> general	14
2.10. <i>Backpropagation</i>	15
2.11. Capa de normalización	15
2.12. Capas totalmente conectadas	15
2.13. VGGNet y VGG16	16

3. Estado del Arte	17
3.1. Métodos de inspección en líneas eléctricas	18
3.2. Fuentes de datos	19
3.3. Detección de componentes	20
3.4. Diagnóstico de fallos	20
3.5. Limitaciones de estudios actuales	20
3.6. Métodos de análisis con <i>Deep learning</i>	21
3.7. El uso directo de frameworks existentes	21
3.8. Extracción de características profundas para clasificación o detección	23
3.9. El problema de la escasez de datos	23
3.10. El mejoramiento de aprendizaje profundo basado en el dominio del conocimiento de inspección de líneas eléctricas	24
4. Metodología de Solución	26
4.1. Generación de la base de datos	26
4.2. Pre procesamiento de imágenes	27
4.2.1. Segmentación manual	27
4.2.2. Recortado y ajuste de tamaño	27
4.3. Entrenamiento	28
4.4. Métricas de evaluación	31
5. Resultados	32
5.1. Caso 1: sintéticas - sintéticas	33
5.2. Caso 2: sintéticas - reales	34
5.3. Caso 3: reales - reales	35
5.4. Caso 4: combinadas - reales	37
5.5. Análisis de todos los resultados obtenidos	38
6. Conclusiones	39
7. Referencias	40

Capítulo 1

Introducción

En los últimos años se ha visto un aumento en el uso de drones para inspección visual, esto dado que pueden tener una mayor facilidad de acceso a ciertos lugares comparado con las inspecciones manuales realizadas por personas.

Esto con la finalidad de localizar y aislar objetos de interés y poder analizarlos de una mejor manera, aumentando así la precisión de detección de fallas o de componentes faltantes, el cual es el caso de los aisladores eléctricos.

1.1. Descripción del Problema

La Comisión Federal de Electricidad (CFE) tiene un gran problema actualmente, las inspecciones que se hacen a todos los elementos que integran las subestaciones eléctricas, son realizados por personas. Esto provoca que no se puedan inspeccionar tan frecuentemente como se quisiera dado que el costo es alto y la tasa de accidentes de los trabajadores incrementa considerablemente, además de que cada inspección toma un largo tiempo. Todo esto aumenta la probabilidad de que haya cortes eléctricos recurrentes y, por lo tanto, elevados costos de mantenimiento. Entonces se propone crear un elemento más que ayude a facilitar estos procesos de inspección.

1.2. Justificación

Es importante resolver este problema porque un sistema que logre segmentar los componentes de las subestaciones eléctricas para después visualizarlos fácilmente sin necesidad de realizar una inspección manual da pie a sistemas autónomos más complejos que detecten fallas o problemas con los componentes de las subestaciones eléctricas en tiempo real. Esto puede reducir en gran medida la tasa de accidentes, así como acelerar el tiempo en el que se llevan a cabo las inspecciones. Además de que la manera en la que se aproximará el problema hace que no se necesiten cientos de imágenes para lograr dichas mejoras, lo cual facilita la recolección de imágenes. Especialmente, los aisladores son de gran importancia porque, cuando fallan, son la principal causa de daños por

fuego o explosiones dentro de hogares y negocios, dado que su propósito es que la energía eléctrica pase a través de ellos, cuidar su estado y monitorearlos reduce el riesgo de ese tipo de accidentes, así como de muertes por descarga eléctrica. Además, son los encargados de suministrar la energía requerida por otros dispositivos como transformadores y subestaciones de control, esto con el fin de protegerlos de alto voltaje. Un buen cuidado y chequeo de los aisladores puede reducir los costos de reparación, en caso de que se pueda prever un fallo con el suficiente tiempo, así como la una disminución de accidentes potencialmente mortales, tanto en hogares como en establecimientos.

1.3. Hipótesis

Es posible entrenar una red neuronal convolucional con pocas imágenes para realizar la segmentación de objetos como cadenas de aisladores eléctricos.

1.4. Objetivo general

Segmentar cadenas de aisladores en imágenes a través de técnicas de aprendizaje profundo considerando pocos datos de entrenamiento.

1.5. Objetivos específicos

- Obtener un conjunto de imágenes virtuales y reales, que contengan cadenas de aisladores para poder realizar su segmentación.
- Analizar e implementar una Red Neuronal Convolucional de tipo Encoder-Decoder para segmentar cadenas de aisladores en imágenes.
- Evaluar si el entrenamiento de la red utilizando imágenes sintéticas puede ser una buena opción para la predicción de la segmentación en imágenes reales.

1.6. Alcances

- Se puede segmentar a nivel píxel los aisladores eléctricos.
- La red neuronal convolucional puede realizar una segmentación usando pocas imágenes durante su entrenamiento.
- Se consideran casos de imágenes donde los objetos tengan oclusiones parciales sin que se afecte en gran medida la precisión.

1.7. Limitaciones

- La entrada del sistema es una imagen donde se enfoque el objeto de interés, aunque en el fondo puede tener otros elementos.
- Se trabaja con solo un tipo de objetos.
- El sistema no funciona en tiempo real.
- La precisión es determinada por la cantidad de imágenes que se utilicen en la etapa de entrenamiento.

Capítulo 2

Marco Teórico

Siendo una clase de los algoritmos de Aprendizaje de Máquina (Machine Learning), el Aprendizaje Profundo utiliza múltiples capas para progresivamente extraer características de mayor nivel de una entrada sin procesar. Un ejemplo de esto es el procesamiento digital de imágenes, en el cual las capas inferiores pueden detectar bordes, mientras que las capas superiores pueden identificar características más complejas y relevantes para los humanos, tales como caras, letras, dígitos, entre otros objetos [1].

El Aprendizaje Profundo ha sido un avance de gran impacto desde principios del año 2000 y su aplicación industrial llegó tan solo 10 años después para problemas de reconocimiento de lenguaje a gran escala. Gracias a esto, la mayoría de los modelos de aprendizaje profundo están basados en las redes neuronales artificiales, de hecho, específicamente en redes neuronales convolucionales [2].

Cada nivel en Aprendizaje Profundo transforma una entrada de información en una representación más abstracta y compuesta. Si se aplica a la tarea de reconocimiento de imágenes, la entrada puede ser una matriz de píxeles sin procesar, en la primera capa de representación de la red se pueden abstraer dichos píxeles y codificarlos para identificar esquinas, en la siguiente capa se pueden localizar bordes, en la tercera se pueden reconocer características más complejas como por ejemplo la nariz y los ojos, y la cuarta capa puede reconocer finalmente que la imagen contiene una cara. Un algoritmo de Aprendizaje Profundo puede aprender por sí mismo a colocar las características de manera óptima en el nivel que les corresponde. Aunque evidentemente, no elimina del todo los ajustes manuales que se deben de hacer en ciertos casos [3], [4].

2.1. ¿Qué es una red neuronal artificial?

Las redes neuronales artificiales son una colección de unidades conectadas, llamadas neuronas artificiales, dado que tienen cierta similitud a las neuronas en los cerebros biológicos. Cada conexión, al igual que las neuronas, al recibir una señal, éstas mandan señales a las neuronas conectadas.

Computacionalmente estas señales, son números reales y cada salida de las neuronas es generada por una función no lineal de la suma de sus señales entrantes. Las conexiones se llaman aristas y estas casi siempre tienen pesos que ayudan a ajustar el aprendizaje que se generará como salida. Las neuronas se colocan en capas, donde cada capa realiza diferentes transformaciones a las entradas, ver figura 1.

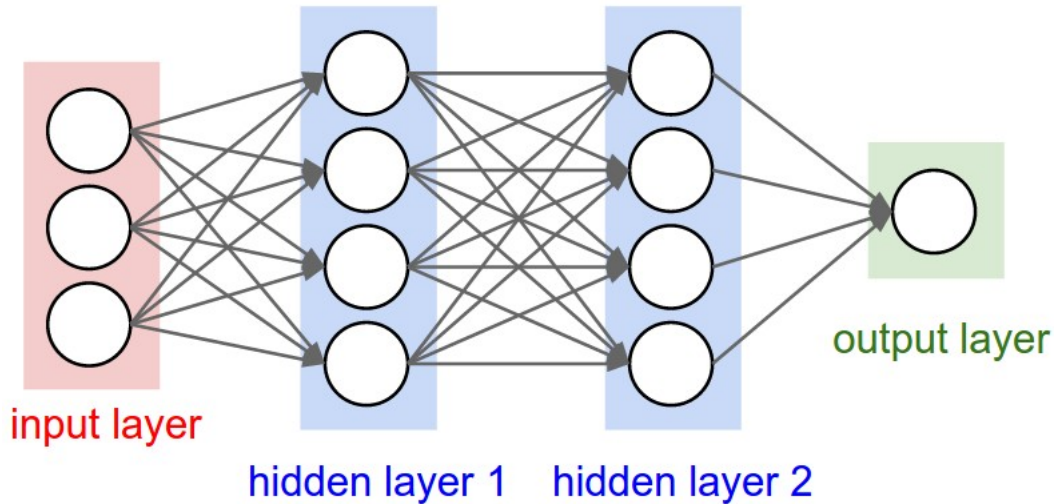


Figura 1. Red neuronal regular de 3 capas [16]

Las redes neuronales artificiales aprenden mediante el procesamiento de ejemplos, los cuales contienen una entrada de la cual se conoce la salida o el resultado esperado, lo cual provoca que se le asocien estas dos partes con cierta probabilidad. Normalmente el aprendizaje se dirige mediante el error que se genera al obtener el resultado de la red neuronal y compararlo con el resultado objetivo. Dado el error, la red neuronal debe de corregir y hacer ajustes a las reglas de aprendizaje aplicadas para minimizar dicho error. Después de cierto número de ajustes, la red neuronal debe de llegar a un punto que cumpla con los criterios predeterminados, para poder terminar el aprendizaje. A esto se le conoce como aprendizaje supervisado. Estos sistemas aprenden a realizar ciertas acciones al considerar los ejemplos proporcionados, generalmente sin ser programados con reglas específicas. Un ejemplo de esto es el reconocimiento de imágenes, dadas ciertas imágenes de entrada la red neuronal es capaz de aprender cuáles características son las que diferencian las imágenes de ejemplo de otras imágenes, para así al momento de recibir una nueva imagen pueda la red neuronal decidir si es similar a las imágenes aprendidas o no.

2.2. ¿Qué es una red neuronal convolucional?

Las redes neuronales convolucionales son una clase de redes neuronales profundas, comúnmente utilizadas para reconocimiento de imágenes. Tienen una o más capas convolucionales que están completamente conectadas o conectadas mediante una técnica llamada *pooling*. Estas son versiones regularizadas de perceptrones multicapa. Los perceptrones multicapa son redes completamente

conectadas, es decir, que cada neurona de cada capa está conectada a todas las neuronas de la siguiente capa. Esta conexión completa hace que las redes tiendan a hacer sobreajustes a los datos. Una forma típica para regularizar estos problemas es mediante las variaciones de los pesos y la minimización de la función de pérdida mientras se recorta aleatoriamente la conectividad. Las redes neuronales convolucionales utilizan relativamente poco pre-procesamiento comparadas con otros algoritmos de clasificación de imágenes, esto es porque las redes neuronales convolucionales aprenden a optimizar los filtros o *kernels* de convolución, que en algoritmos tradicionales están diseñados por el usuario manualmente. La independencia de conocimiento anterior y de intervención humana es una gran ventaja para estas redes neuronales.

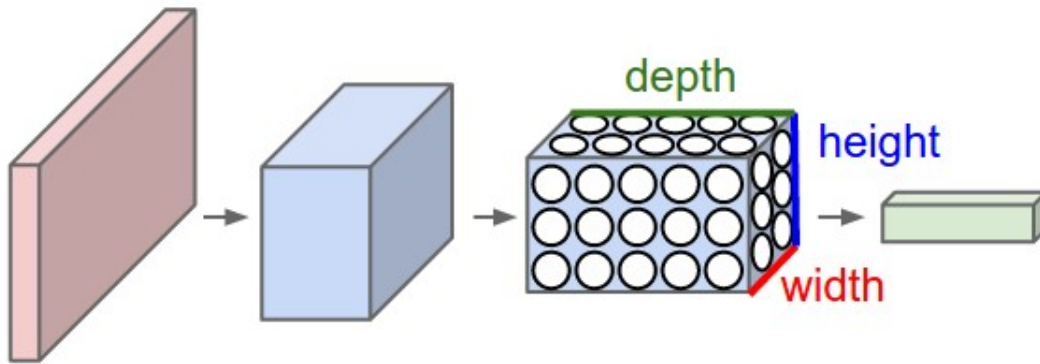


Figura 2. Ejemplo de convolución de una red neuronal convolucional, la capa roja es la imagen de entrada, la azul es el resultado tridimensional de una convolución y la última capa es la salida aplanada lista para que se le aplique una función de activación. [16]

2.3. Funciones de activación

Las funciones de activación, es la manera en la que se obtienen las salidas de cada nodo. Estas funciones de activación se pueden dividir en 2 grupos: funciones de activación lineales y no lineales. Las funciones de activación lineales, como son representadas con una línea, sus salidas no están confinadas a un rango determinado. Esto no ayuda en el uso de redes neuronales dado a los parámetros o complejidad de los datos que se les da de entrada. Mientras que las no lineales facilitan que el modelo generalice o se adapte a una gran variedad de datos y a diferenciar las salidas. Algunas de las funciones de activación son:

1. Función sigmoide o logística: La razón principal por la que se utiliza esta función de activación es porque queremos que las salidas solo se encuentren entre 0 y 1. Esto hace que se utilice en modelos para predecir la probabilidad de una salida. En la figura 3 se puede observar la gráfica general de la función.

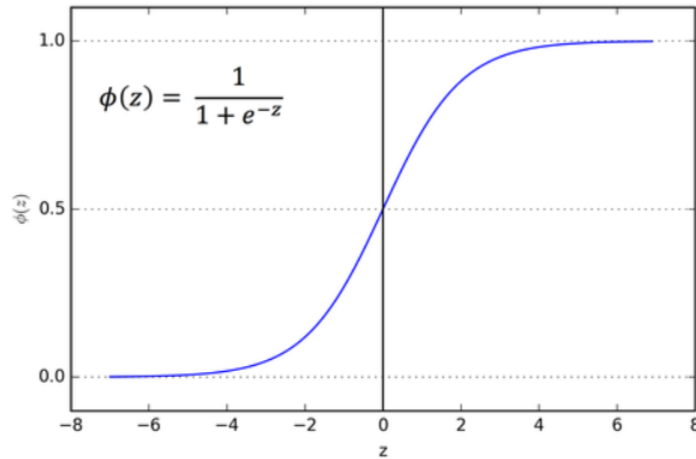


Figura 3. Gráfica de la función sigmoideal. Obtenida de [41]

2. Función Tanh o tangente hiperbólica: El rango de la función tanh va de $(-1, 1)$ y tiene forma de S al igual que la sigmoideal. Una ventaja de esta función de activación es que las salidas negativas se ven fuertemente mapeadas como negativas y los ceros se grafican mucho más cerca del 0 comparado con la sigmoideal, tal como se puede ver en la figura 4. Usualmente esta función se utiliza en clasificación entre dos clases.

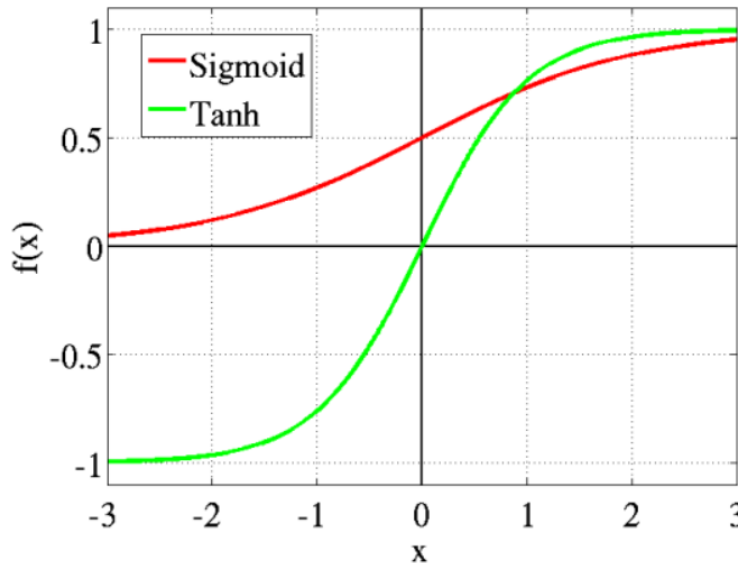


Figura 4. Gráfica de comparación de la función sigmoideal y tanh [41]

3. Función ReLU o Rectified Linear Unit: Es la función de activación más usada en el mundo actualmente, dado que es la que se utiliza en casi todas las redes neuronales convolucionales y en aprendizaje profundo. La gráfica de esta función mapea $f(z)$ a cero cuando z es menor que cero y $f(z)$ es mapeada a z cuando z es mayor o igual a cero. Su rango va de $[0, \infty)$. Un detalle importante es que, dado que los valores negativos son mapeados a cero inmediatamente, tiende a ser una función que no es tan apropiada para datos negativos. Esto se observa en la figura 5.

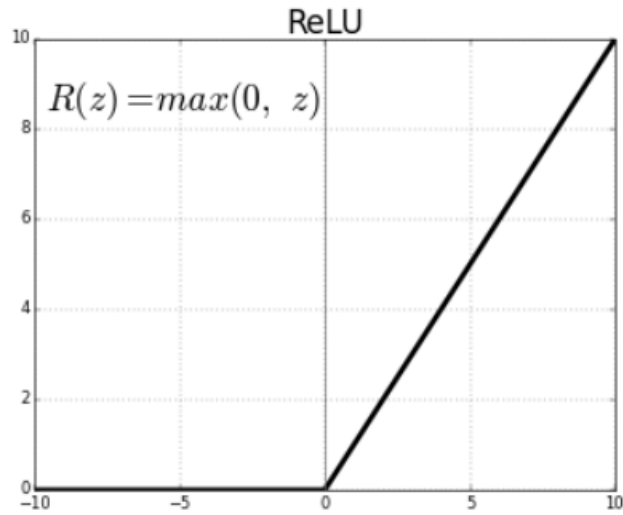


Figura 5. Gráfica de la función ReLU. [41]

4. Función Leaky ReLU: Esta función de activación intenta resolver el problema de dying ReLU, que sucede cuando las salidas comienzan a ser 0 dado valores anteriores negativos y eso hace que las neuronas “mueran”, es decir, que siempre tengan salidas 0 y no aporten nada al resto de la red. La fuga o *leak* se obtiene al multiplicar las salidas de la función ReLU en valores negativos por un factor a , lo que hace que el rango de dicha función sea ahora $(-\infty, \infty)$. Usualmente el valor otorgado a a es 0.01. La gráfica de dicha función se observa en la figura 6.

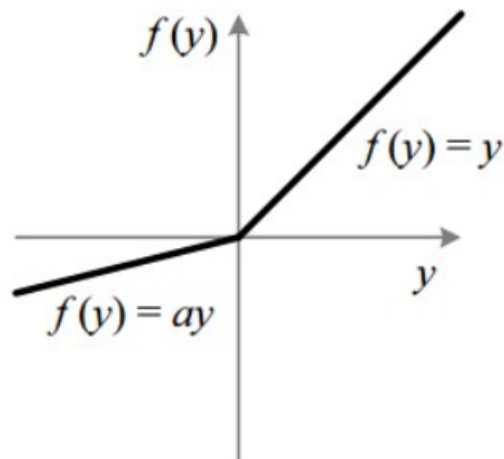


Figura 6. Gráfica de la función *Leaky ReLU* [41]

2.4. Segmentación de imágenes

La segmentación de imágenes es un proceso de visión por computadora en el que se particiona la imagen digital en varios segmentos o conjuntos de píxeles. El objetivo de este proceso es el de simplificar y/o cambiar la representación de una imagen de tal manera que sea información más significativa y fácil de analizar [9] [10]. Usualmente se utiliza para localizar objetos y fronteras (líneas o curvas) en las imágenes. Este proceso intenta asignar una etiqueta a cada píxel de la imagen, tal que píxeles con la misma etiqueta compartan características. Estas características o propiedades calculadas como el color, intensidad o textura son similares con las que comparten región en la imagen y a su vez, estas regiones o segmentos de imágenes hacen que conjuntamente se cubra por completo la imagen de la entrada inicial o un conjunto de contornos extraídos de dicha imagen [9]. La segmentación de imágenes se utiliza comúnmente en representaciones médicas por medio de imágenes, en donde se aplica a un grupo de imágenes para que los contornos resultantes puedan ser utilizados para reconstrucciones 3D con ayuda de algoritmos de interpolación [11]. Otras aplicaciones de la segmentación de imágenes son:

- Recuperación de imágenes basadas en contenido.
- Detección de objetos (transeúntes, rostros, luz de frenos en automóviles, etc.).
- Tareas de reconocimiento (reconocimiento facial, reconocimiento de huellas digitales o reconocimiento del iris).
- Sistemas de control del tráfico.
- Vigilancia por medio de vídeos.

Existen dos maneras de aplicar la segmentación de imágenes, la segmentación por instancias y la segmentación semántica. Estos tipos de segmentación se pueden observar en la figura 7.

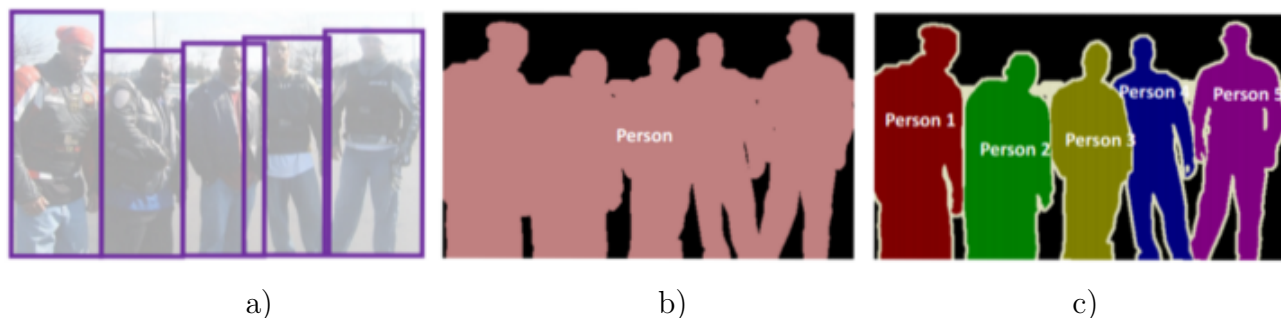


Figura 7. Tipos de segmentación de imágenes. a) Detección de objetos. b) Segmentación semántica. c) Segmentación por instancia. [44]

La segmentación por instancias se refiere a que se identifica una instancia del objeto para cada píxel en la imagen. Un ejemplo de esta segmentación es cuando se detecta a cada persona en una imagen que contiene un grupo de personas como un objeto individual [12]. La segmentación

semántica identifica la clase de cada objeto que se detecta píxel a píxel. Esto se puede ejemplificar cuando la imagen solamente se divide de forma que se pueden observar las diferentes clases de objetos existentes en la imagen, pero no hay información sobre el número de instancias por clase [13]. Se puede conceptualizar como un conjunto de capas en las que cada capa asigna un color C a cada píxel que cumpla con el requisito de ser parte de la clase K y a los demás píxeles se les asigna el color negro. Entonces al combinar dichas capas, se obtiene una imagen completamente cubierta por colores que identifican la clase o grupo semántico de los objetos en la imagen de entrada.

2.5. Definición desde el punto de vista intuitivo

Sin tomar el punto de vista de analogías de cerebros o neuronas, los parámetros de las capas convolucionales consisten en un conjunto de filtros con la habilidad de aprendizaje. Cada filtro ocupa un espacio pequeño, pero se extiende a través de una profundidad completa del volumen de la entrada. Un ejemplo de esto es, la primera capa de un filtro típico, la cual tiene un tamaño de $5 * 5 * 3$. Durante el proceso hacia adelante, se realiza la convolución en cada filtro a lo largo de su anchura y altura del volumen de la entrada y se calcula el producto punto de los registros del filtro y la entrada en cualquier posición. Mientras se realiza la convolución sobre el ancho y la altura del volumen de la entrada se produce un mapa de activación bidimensional que nos proporciona las respuestas de dicho filtro y la entrada en cualquier posición. Intuitivamente, la red aprenderá los filtros que activan cuando observan cierto tipo de característica visual tal como una orilla de una orientación en específico o una mancha de cierto color en la primera capa, o eventualmente todo el panal o los patrones con figura de llanta en capas superiores de la red. Ahora se hará un conjunto de filtros en cada capa convolucional y cada uno de ellos producirá un mapa de activación bidimensional. Se apilarán estos mapas de activación a lo largo de la dimensión de profundidad y producir así el volumen de la salida.

2.6. Acomodo espacial

Esto se refiere a cómo se acomodan las neuronas contenidas en el volumen de la salida, así como cuántas neuronas se encuentran en dicho volumen de salida. Se utilizan tres hiper parámetros para controlar el tamaño del volumen salida: la profundidad, el *stride* y el *zero-padding*. Estos se pueden definir de la siguiente manera:

1. Profundidad: Este hiper parámetro corresponde al número de filtros que se usarán, cada uno de ellos aprenderá a buscar algo diferente de la entrada. Por ejemplo, si la primera capa convolucional tiene una entrada de una imagen sin procesar, se pueden activar diferentes neuronas a lo largo de la dimensión de profundidad dada la presencia de varias orientaciones de aristas o manchas de color. Usualmente, al conjunto de neuronas que buscan en la misma región se le conoce como columna profundidad o *fibre*.
2. En cuanto al *stride*, cuanto más grande sea su valor, el volumen de salida será cada vez más pequeño dado que al aumentar este valor a n , estamos saltando $n-1$ píxeles.

- Algunas veces el tamaño del *zero-padding* es un hiper parámetro dado que un buen tamaño nos permitirá controlar el tamaño espacial de los volúmenes de salida. Comúnmente se usa para preservar el tamaño espacial de la entrada, así que el ancho y alto de la entrada y salida serán iguales.

2.7. Arquitectura *encoder-decoder*

Las redes neuronales convolucionales con arquitectura *encoder-decoder* son utilizadas para trabajos en los que se necesitan predicciones de una gran cantidad de píxeles como en la segmentación semántica [5], determinación del flujo óptico y mapas de disparidad [6], así como detección de contornos [7]. La parte del codificador (*encoder*) es la que se encarga de calcular características abstractas de manera progresiva donde las características van siendo cada vez de mayor nivel, ya que los campos receptivos del codificador se incrementan conforme más profundidad tenga dicho codificador. También la resolución espacial de los mapas de características se reduce poco a poco mediante una operación de *down-sampling*, mientras tanto, el decodificador (*decoder*) calcula los mapas de características que van incrementando vía *up-sampling*. Esta red tiene la capacidad de modelar características como forma o apariencia, así como también puede modelar relaciones espaciales de gran alcance. Este atributo de modelar características locales y globales hace que sea apta para aprender gráficos semánticos [8]. En la figura 8 se muestra una representación de una arquitectura *encoder-decoder*.

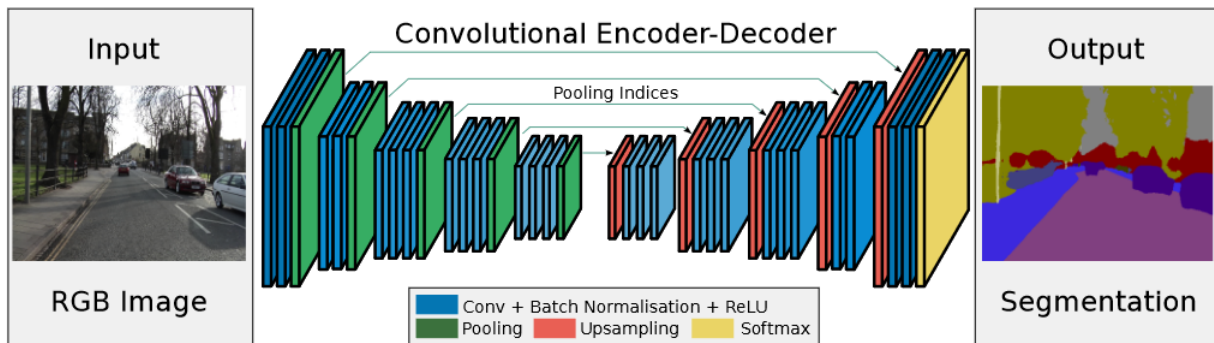


Figura 8. Una representación de una la arquitectura *encoder-decoder* de SegNet. Obtenida de [5]

2.8. Capa de *pooling*

Es común que periódicamente se inserten capas de *pooling* entre capas convolucionales inmediatas en una arquitectura convolucional. La función de esta capa es de reducir el tamaño de la representación progresivamente para conseguir una menor cantidad de parámetros y cálculo en la red, además de controlar el *overfitting* en el proceso. La capa de *pooling* opera independientemente en cada fragmento de profundidad de la entrada y le cambia el tamaño espacialmente utilizando la operación MAX. La forma más usual de esta capa es con filtros de $2 * 2$ aplicada con un *stride* de 2 *downsamples* por cada fragmento de profundidad en la entrada por 2 a lo largo del ancho y el

alto, esto hace que se descarte el 75% de las activaciones. Cada operación MAX toma el máximo de entre 4 números. La dimensión de profundidad queda sin cambios. Es importante saber que las capas de *pooling* demasiado grandes pueden ser muy destructivas. Un ejemplo básico de *pooling* se puede ver en la figura 9.

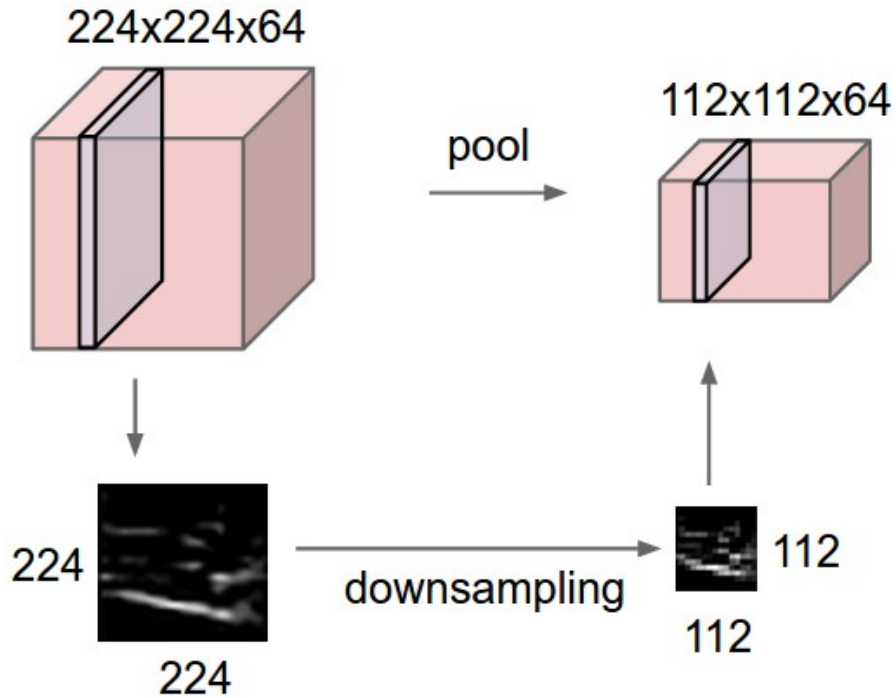


Figura 9. Ejemplo de un pooling de 2×2 con stride de 1 y cómo afecta este proceso de *downsampling* a la imagen que se le aplicó. Obtenida de [16]

2.9. Pooling general

Además del *max pooling*, las unidades de *pooling* pueden realizar otras funciones como *average pooling* o también *L2-norm pooling*. *Average pooling* se utilizaba históricamente, pero recientemente ha tenido menor uso comparado con la operación de *max pooling*, la cual ha mostrado tener mejor desempeño en la práctica. En la figura 10 se muestra un ejemplo de *max pooling*.

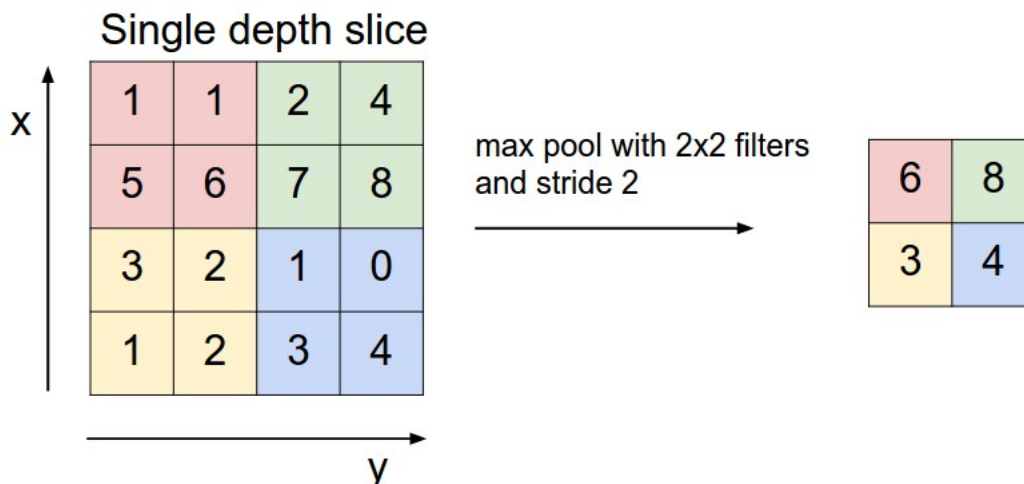


Figura 10. Ejemplo de *max pooling* de $2 * 2$ con una *stride* de 2 [16]

2.10. *Backpropagation*

El *recall* de la *backpropagation* que se hace en un pase hacia atrás por una operación *max* tiene una interpretación simple dado que solamente se toma la ruta del gradiente a la entrada que tenga el mayor valor en el pase hacia adelante. Por lo tanto, durante el pase hacia adelante de la capa de *pooling* es común mantener un registro del índice de la activación *max* para que la ruta del gradiente sea eficiente durante la *backpropagation*.

2.11. Capa de normalización

Se han propuesto varios tipos de capas de normalización para las arquitecturas convolucionales, estas propuestas intentan implementar esquemas observados en cerebros biológicos. Aunque, estas capas han decaído en uso dado que, en práctica, la contribución ha mostrado ser mínima si es que hay alguna.

2.12. Capas totalmente conectadas

Las neuronas de las capas totalmente conectadas tienen conexiones a todas las activaciones de las capas anteriores como en las redes neuronales regulares. Sus activaciones pueden ser calculadas con una multiplicación de matrices seguida de un desface contra *bias*.

2.13. VGGNet y VGG16

VGGNet, fue una red que compitió en la ILSVRC 2014, desarrollada por Karen Simoyan y Andrew Zisserman, fue conocida como VGGNet. La contribución que hizo esta red fue mostrar que la profundidad de la red es un componente crítico para conseguir un buen desempeño. Su mejor red contenía 16 capas convolucionales/totalmente conectadas y contiene una arquitectura extremadamente homogénea que solo realiza convoluciones de $3 * 3$ con un pooling de $2 * 2$ desde el principio hasta el final. Una desventaja de la VGGNet es que es mucho más costosa de evaluar y usa mucha más memoria y parámetros (140 millones). La mayoría de estos parámetros están en la primera capa totalmente conectada, desde este momento se encontró que las capas totalmente conectadas se pueden quitar sin perder desempeño, esto reduce el número de parámetros necesarios.

Capítulo 3

Estado del Arte

El crecimiento de la popularidad de los vehículos aéreos no tripulados hace posible que se puedan recopilar grandes cantidades de datos sobre inspección de líneas eléctricas y surgen problemas urgentes a los que se les quiere dar solución, problemas como la veracidad de este tipo de inspecciones sin personas directamente realizándolas, así como la seguridad y la integridad de la transmisión de energía eléctrica.

Pocas bases de datos se han creado con el objetivo de realizar segmentación *few-shot* con ellas, en [42] se buscó crear una llamada FSS-1000. Esta base de datos está compuesta por 1000 clases y cada clase contiene 10 imágenes, haciendo que para incluir una clase nueva simplemente se necesiten 10 imágenes con anotación a nivel pixel. Esta base de datos contiene 584 clases compartidas con la base de datos ILSVRC y tomando en cuenta que los autores de FSS-1000 consideraron que esa base de datos se enfocaba mucho en animales decidieron introducir objetos cotidianos para conseguir que al entrenar una red con dicha base de datos se tenga una mayor cantidad de objetos artificiales creados por el humano, así como objetos naturales y orgánicos. El tamaño de sus imágenes se fijó en $224 * 224$ pixeles y las imágenes que tuvieran una razón de tamaño menor a 0.5 o mayor a 2 fueron eliminadas dado que podrían causar problemas por el gran re escalamiento de las imágenes, haciendo que potencialmente perdieran características importantes para el aprendizaje.

La inspección de las líneas eléctricas surge como una necesidad para poder obtener un suministro ininterrumpido de la energía eléctrica, el cual es un tema sumamente importante debido a la dependencia creciente a dicha energía en las sociedades actuales. Una línea eléctrica está compuesta de diferentes tipos de componentes con distintas funciones, como los aisladores, las torres eléctricas y los conductores. Dado que estas inspecciones se realizan al aire libre, en terrenos complejos y con clima constantemente en cambio, los componentes de una línea eléctrica pueden ser dañados frecuentemente. Un solo componente dañado puede causar un corte eléctrico, aunque usualmente estos cortes ocurren cuando varios componentes están dañados. Un corte local puede ser relativamente barato de arreglar, pero estos cortes pueden incluso causar problemas mayores dependiendo de la gravedad de la falla de los componentes, posiblemente llegando a cortes eléctricos fuera de la localidad o región del corte inicial.

3.1. Métodos de inspección en líneas eléctricas

Usualmente estos métodos de inspección utilizan inspección tanto aérea como terrestre. Ambos métodos identifican la condición de dichas líneas eléctricas utilizando observación visual. La parte terrestre se lleva a cabo por un equipo que viaja a través de los corredores de las líneas eléctricas a pie o en un vehículo fuera de la carretera. Durante esta inspección terrestre, normalmente el equipo utiliza herramientas de observación como binoculares y cámaras infrarrojas. A pesar de que este método terrestre se ha llevado a cabo durante décadas por su precisión, cuenta con algunos problemas como son: actividades intensivas, con baja eficiencia y los terrenos extremadamente complejos hacen que gradualmente se esté buscando reemplazar por el método aéreo. En el caso del método aéreo, usualmente se realiza mediante un sistema de escalado o un sistema aéreo. En este último se utiliza un robot móvil para atravesar obstáculos que se encuentran en las líneas eléctricas e inspeccionar los componentes que van pasando durante su trayectoria. Los robots que escalan pueden obtener imágenes de muy alta calidad dado su proximidad con los conductores. A pesar de esto, las desventajas del sistema escalador limitan su uso incluyendo los daños a las líneas eléctricas, baja eficacia, inspecciones incompletas, obstrucción por obstáculos y el alto costo que pueden tener por la gran cantidad de instrumentos y personal necesarios. La inspección de los componentes de líneas eléctricas es un tema de investigación muy importante en ese ámbito. Esta inspección tiene el objetivo de identificar las condiciones en las que se encuentran dichos componentes. Los componentes más comunes para investigación son las torres eléctricas, los conductores y los aisladores eléctricos, así como los *electrical fittings*.



a)



b)

Figura 11. a) Inspección de una torre eléctrica con ayuda de un dron aéreo. b) Helicóptero realizando inspección de una torre eléctrica. Obtenidas de [43]

En el caso del sistema aéreo, se inspecciona las líneas electrónicas utilizando vehículos aéreos como helicópteros, *UAV* (del inglés *Unmanned Aerial Vehicle*) multi rotor y *UAV* con alas fijas. Algunas de las ventajas para el método aéreo son:

1. Acceso a localizaciones difíciles de acceder, lo que significa que tienen una gran flexibilidad para adquirir datos.
2. Capacidad de cargar múltiples instrumentos de sensores para la inspección.

-
3. Responden al problema de baja eficacia y daño a las líneas eléctricas que muestran los sistemas escaladores.

Dado que los UAV de multi-rotor pueden volar relativamente cerca de las líneas eléctricas y que son mucho más baratos que otros vehículos aéreos con bajo costo por uso, se ha convertido en el método más común para la inspección de líneas eléctricas.

3.2. Fuentes de datos

Los datos obtenidos mediante los diferentes métodos de inspección se deben de analizar, ya sea por humanos o por computadoras para identificar la condición en la que se encuentran las líneas eléctricas. Los diferentes tipos de datos tienen distintas maneras de analizarse. Por lo tanto, es importante determinar qué tipo de fuentes de datos se está utilizando en el sistema de inspección. Se pueden considerar dos categorías: datos en forma de imagen y datos sin forma de imagen. Los datos sin forma de imagen se refieren a datos que son obtenidos mediante un escáner láser aéreo, estos datos también se conocen como datos georreferenciados en la nube. Estos dispositivos pueden generar datos 3D muy detallados con ayuda de la coordinación de la información de los objetos y se puede aplicar al mapeo y reconstrucción 3D de los corredores de una línea eléctrica. Los datos en forma de imagen forman la mayor fuente de datos obtenidos de los diferentes métodos de inspección porque la mayoría de las condiciones de una línea eléctrica pueden ser identificadas a través de observación visual. Estos datos incluyen en su mayoría, imágenes visibles, imágenes infrarrojas, imágenes ultravioletas, imágenes de apertura sintética de radar e imágenes de satélites ópticos. Cada uno de estos tipos de imágenes tienen su uso particular, por ejemplo, las imágenes infrarrojas reflejan la temperatura de los objetos tal que se pueda utilizar dicha información para detectar calor fuera de lo normal. Las imágenes ultravioletas se utilizan usualmente para detectar descargas en forma de corona en las líneas eléctricas. Las imágenes de apertura sintética de radar ofrecen cobertura de una gran área, que han sido utilizadas en el monitoreo de la vegetación cercana a las líneas eléctricas. El uso de imágenes tiene varias ventajas como:

1. La gran mayoría de las fallas tienen características visibles que pueden ser identificadas mediante observación visual.
2. Tener una imagen visible es más apropiada para ser intuitivamente utilizada por los humanos.
3. La adquisición de las imágenes visibles es bastante flexible, de bajo costo y alta calidad que se beneficia de las cámaras con un desarrollo muy bueno y también se beneficia de la tecnología de fotografía aérea.

Los análisis que se llevan a cabo en estas inspecciones se realizan desde dos enfoques diferentes, detección de componentes y el diagnóstico de fallas. Ambas son de gran importancia. La detección de componentes es importante dado que se necesita para la identificación de futuros fallos, pero también puede ser utilizado para otros fines como navegación UAV, administración de recursos y rastreo por video. Las investigaciones para detección de componentes se dividen en cinco grupos

de acuerdo con las características de las imágenes que se utilizan, estas son: color, forma, textura, fusión y profundidad. El diagnóstico de fallas determina la condición de las líneas eléctricas. La manera en que se enfoca este diagnóstico es dependiente de la perspectiva de diferentes fallos como fallos de superficie y tapas faltantes de aisladores, corrosión de las torres, nidos de pájaros, conductores rotos, cuerpos externos, invasión de vegetación, *fittings* rotos y faltantes, entre otros.

3.3. Detección de componentes

La detección de componentes es un prerrequisito esencial para cualquier análisis posterior y esto ha causado que las investigaciones referentes a esta detección se hayan incrementado en los últimos años. Este procedimiento se puede dividir en dos fases: la extracción de características y la clasificación de características.

3.4. Diagnóstico de fallos

Se considera el diagnóstico de los fallos de los componentes de las líneas eléctricas cuando se utilizan imágenes de inspección visibles. Los estudios que rodean este diagnóstico son muchos menos comparado con los que hay sobre detección de componentes dadas las siguientes razones:

1. Los componentes con fallos, aunque hacen daño a los sistemas eléctricos, son mucho menos comunes comparados con la falta de dichos componentes.
2. Hay múltiples tipos de fallos en un mismo componente.
3. Existen varias manifestaciones del mismo tipo de fallo en las imágenes.

Estas razones causan que exista una falta de información y esto limita el uso de aproximaciones usando aprendizaje, en cambio, los métodos manuales son difíciles de manejar dada la extensa variedad de fallos. Usualmente el procedimiento del diagnóstico de fallos se compone de dos fases: detectar el componente e identificar la falla. En la primera fase se requiere una “Región de interés”, la cual debe de ser detectada y recortada para poder deshacerse del fondo para análisis posteriores. En la segunda fase el método de identificación de fallas puede ser aplicado en la “Región de interés”. En algunos estudios no se considera la primera fase dado que el componente ya está en condiciones para realizar la segunda fase. Existen ciertas fallas que no necesitan necesariamente de una detección de componentes tales como nidos de aves y cuerpos externos.

3.5. Limitaciones de estudios actuales

Dos limitaciones importantes que se han encontrado en investigaciones anteriores son la falta de investigación de ciertos componentes y sus fallas, la otra es la falta de ingeniería práctica. Con

respecto a la primera limitación, se ha podido observar que la mayoría de las investigaciones se han hecho sobre los aisladores y sus fallas, a los otros componentes se les ha prestado poca atención. Esto sucede dado a la facilidad de identificar los aisladores por la forma estandarizada que se les ha dado, esto significa que tienen pocas variaciones en las imágenes, haciendo que los algoritmos que las identifican sean más fáciles. Otra causa de este fenómeno es su tamaño moderado, un problema que enfrentan las torres eléctricas es que son demasiado grandes y los conductores son demasiado delgados. Estas características dificultan la identificación de dichos objetos. Por otro lado, la segunda limitación que se encontró en [43] es la falta de pruebas que se han realizado con ingeniería real. En un entorno controlado y cerrado en donde se tienen los datos e imágenes de los fallos para distribuirlas y tener conjuntos de entrenamiento y de prueba con buenos números, aunque usualmente esto no se puede garantizar en un entorno real, en el que ciertas fallas no se han considerado con tanta frecuencia como otras. Además de que los fondos varían de región a región, haciendo la identificación de fallos y de los mismos objetos más complicada de generalizar. Otro factor que limita la practicidad de la aplicación de la inspección es el costo computacional. Dado que se deben de tratar imágenes de alta resolución, el análisis se torna costoso y pesado, cuando existe una limitación en los recursos computacionales, estas inspecciones se vuelven poco viables por la alta eficiencia requerida en el cómputo. Además, el concepto de acelerar estos procesos computacionales no ha sido investigado lo suficiente como para mejorar el tiempo de cómputo, esto añadido a que los métodos de análisis raramente ofrecen estos tiempos en literaturas existentes.

3.6. Métodos de análisis con *Deep learning*

El *deep learning* se ha utilizado ampliamente en detección de automóviles o reconocimiento facial, pero en 2018 la inspección de líneas eléctricas se volvió un tema de importancia para la investigación con respecto al uso de este tipo de aprendizaje máquina. Las formas en las que se exploran las redes neuronales para la inspección de líneas eléctricas actualmente son las siguientes:

3.7. El uso directo de frameworks existentes

Una estructura de redes neuronales comúnmente usada en la inspección de líneas eléctricas con enfoque en fallas de aisladores, detección de torres eléctricas y detección de *fittings* es la *Faster R-CNN*. [30, 17, 18, 19]. Un ejemplo de esto es la aplicación que se hizo en [30] en donde se detectaron los aisladores y las tapas faltantes de manera separada. Probaron el método con imágenes de aisladores en tres niveles diferentes de voltaje y prepararon 1000 ejemplos para el entrenamiento, así como 500 para probar dicho entrenamiento para cada nivel. Ahora, para la falta de tapas se tomaron solamente 120 imágenes de las cuales 80 de ellas fueron para entrenamiento. En el experimento realizado se ajustaron las imágenes para que fueran de $500 * 500$ píxeles y se utilizó *data augmentation*, esto incluyendo recortado y rotaciones de las imágenes para hacer el conjunto de datos más grande.

Bian et al. [17] utilizó la misma estructura *Faster R-CNN* para detección de torres eléctricas. Se utilizaron en total 1300 imágenes aéreas, las cuales fueron preparadas para experimentos y una *10-fold cross-validation* fue aplicada para encontrar el mejor modelo. Hui et al [18] usó también la

misma estructura para localizar torres eléctricas. Más adelante se utilizaron redes neuronales totalmente conectadas (FCNs) para extraer los conductores de las imágenes. Los datos utilizados fueron 1280 imágenes de torres eléctricas, de estas 1000 fueron destinadas para el entrenamiento, para los conductores se destinaron 600 imágenes con 400 de estas siendo utilizadas para entrenamiento. Wang et al. [19] aplicó la misma estructura para detectar los *fittings* los cuales son utilizados para conectar los conductos que hacen más fácil llevar los cables eléctricos a través de estructuras, esto incluyendo los de tipo espaciales, *dampers*, y anillo arqueado. Para cada tipo de *fittings* se usaron 1500 ejemplos para el entrenamiento y 500 imágenes fueron preparadas para probar dicho entrenamiento, todas las imágenes fueron re escaladas a un tamaño de $500 * 500$ pixeles. Para obtener una alta velocidad de cómputo, se utilizó una estructura en la fase de detección como la YOLO (del inglés *You Only Look Once*) [21,22] y la SSD (del inglés *Single Shot Detection*) [20]. Wang et al. [21] utilizó YOLO para detectar aisladores en las imágenes con un *greyscale colorspace*. Su base de datos estaba compuesta de 1000 imágenes con 800 de estas siendo para el entrenamiento, estas imágenes fueron obtenidas de un laboratorio y de líneas eléctricas al aire libre. Todas las imágenes fueron re escaladas a un tamaño de $448 * 448$ para hacerlas coincidir con el tamaño de entrada que requería la red. Chen et al [22] utilizó una estructura YOLO mejorada, a la cual se le denomina YOLOv3, para detectar torres eléctricas. Dado que los datos de los que disponían eran insuficientes, construyeron una base de datos generando las imágenes modificadas. Un total de 13,429 imágenes fueron utilizadas en dicho experimento, de estas 11,951 fueron destinadas a entrenamiento y 1,478 para probar el entrenamiento, la resolución de imágenes que aceptaba la red fue de $352 * 352$ pixeles.

Los autores han discutido que el tamaño de pixeles que necesita como entrada la red fue un factor importante para el desempeño de dicho método. Xu et al [20] propuso un método basado en SSD para la detección de aisladores. En total 2000 imágenes fueron tratadas mediante rotación para la *data augmentation* que se realizó, extendiendo el número a 6000 con 500 para validación y 5,500 para entrenamiento. En los experimentos realizados se utilizó una resolución de $512 * 512$ pixeles, esto mostró una precisión más alta comparado con imágenes de $300 * 300$ pixeles, mientras ambos experimentos obtuvieron el mismo resultado de realizar la detección en tiempo real. Algunos estudios aplicaron algunas estructuras como cGAN poco comunes para la detección de componentes de líneas eléctricas [23, 24]. Chang et al. [23] hizo el reconocimiento de los conductores eléctricos utilizando esta misma estructura. Las imágenes aéreas fueron utilizadas como entrada para la cGAN y la máscara de las imágenes que solo contenían el conductor fueron generadas. Las resoluciones de la entrada y salida fueron de $256 * 256$ y $128 * 128$ respectivamente. Se prepararon tres bases de datos, incluyendo un conjunto de entrenamiento de 5000 imágenes, el conjunto de prueba simple fue de 500 imágenes y el de prueba difícil contaba con 500 imágenes igualmente. Los autores utilizaron cGAN para la detección de aisladores [24]. Se propuso una estrategia de dos fases para obtener una precisión mayor en el modelo. En la primera fase de dicha estrategia, el modelo fue entrenado con los ejemplos de posiciones con una anotación no tan precisa. Después el mismo modelo siguió el entrenamiento, pero esta vez con ejemplos con una anotación muy fina. Se recolectaron 3000 imágenes del Internet para evaluar el modelo. La entrada y salida del modelo cGAN fueron del mismo tamaño de $256 * 256$.

3.8. Extracción de características profundas para clasificación o detección

Algunos ejemplos del uso de la extracción de características profundas para clasificación [25,26] o detección [27] se pueden encontrar en los siguientes estudios. Zhao et al. [25] clasificó la condición de los aisladores mediante AlexNet. Las características fueron extraídas por la red AlexNet sin entrenar, la cual fue pre entrenada con el conjunto de datos de ImageNet. Se utilizaron un total de 1000 imágenes de tamaño $256 * 256$ pixeles, de estas el 70 % fueron utilizadas para entrenamiento y el resto para probar el entrenamiento con el método propuesto. El propósito de los niveles de deterioro se estimó para las torres eléctricas. Maeda et al. [26] extrajo las características visuales con ayuda de LRF (del inglés *Local receptive fields*), el cual hizo una convolución y *pooling* similar a las redes neuronales convolucionales. Esta manera se diferencia de los estudios tradicionales basados en imágenes, dado que combinan las características de texto que fueron extraídas por una capa oculta. Se extrajeron y clasificaron dos tipos de características más adelante, esto fue realizado por DELM (del inglés *Deep Extreme Learning Machine*. En el experimento se usaron 3,107 imágenes de $50 * 50$ pixeles de resolución, a estas se les realizó una validación quíntuple como método de verificación. Yang et al. [27] estableció una red neuronal convolucional de 9 capas para extraer las características profundas de sub-ventanas de la imagen original para la detección de fallos en los aisladores. Un ejemplo de esto es cuando se toma una imagen aérea con una resolución de $1280 * 720$ pixeles y se divide en 15 imágenes más pequeñas por la ventana deslizante adaptable. Luego, la sub-ventana puede ser determinada por la red neuronal convolucional entre dos clases: la normal y la anormal. El modelo de red neuronal convolucional fue entrenado en 2,610 sub-ventanas que se obtuvieron de 205 imágenes sin modificaciones y fue puesta a prueba en 341 imágenes del mundo real.

3.9. El problema de la escasez de datos

La escasez de datos es un problema difícil en el campo de análisis de la inspección de líneas eléctricas. Algunos artículos tratan de realizar síntesis de datos [14] y data augmentation [20, 14]. Los siguientes estudios han realizado investigaciones sobre el problema de escasez de datos. Chang et al. [24] utilizó cGAN para la detección de aisladores, pero dada la dificultad de obtener imágenes aéreas del mundo real propusieron un método para generar imágenes sintéticas de 65 imágenes de aisladores reales. La región del aislador se sobrepone con varias imágenes de fondo con distintos parámetros tales como ruido gaussiano y transparencia. La base de datos sintética incluyó tres categorías de muestras: muestra con aislador, muestra sin aislador y muestra con pseudo objetivos. El modelo cGAN se entrenó con 800 imágenes sintéticas y fue probado con 200 imágenes de aisladores reales. La entrada y salida de dicho modelo tuvo un tamaño de $512 * 512$ pixeles. Tian et al. [19] propuso un método paralelo para resolver el problema de la escasez de diversidad de datos adquiridos para la inspección. La entrada original fue procesada mediante diferentes operaciones como rotación, espejo y eliminación de niebla, luego se le dieron a la red de cascading para el diagnóstico de fallas. Luego se utilizaron como entrada las imágenes paralelas, los resultados paralelos se generarían y por un mecanismo de decisión por votación que fue diseñado para determinar

el resultado final se tomó como el paso final. Además de incrementar la diversidad de los datos, algunos estudios discutieron el uso de aprendizaje de transferencia. Bai et al. [31] determinó la superficie de fallo de los aisladores, esto basado en redes de *Spatial Pyramid Pooling* o SPP-Net, con la transferencia de aprendizaje. En el experimento el modelo fue entrenado primeramente con la base de datos ImageNet, el cual contenía 1.2 millones de muestras para entrenamiento. Luego, el mismo modelo fue entrenado a más profundidad, para realizar los ajustes finos, con una base de datos pequeña sobre fallas de aisladores. Miao et al. [32] introdujo una estrategia de dos fases para realizar los ajustes finos en la red SSD para detectar aisladores. Se prepararon dos tipos de bases de datos de aisladores para el método propuesto: el conjunto básico y el conjunto específico. El primero contenía imágenes aéreas con varios tipos de aisladores sobre diferentes fondos, las cuales se encontraron en gran cantidad. El segundo estaba compuesto de imágenes con un aislador en un fondo específico, por ejemplo, un aislador de porcelana sobre un bosque como fondo, este conjunto tenía pocas imágenes. La implementación de los ajustes finos en la primera fase fue similar a [31]. Pero en lugar de la base de datos de ImageNet y la pequeña base de datos pequeña mencionadas en [31], utilizaron la base de datos COCO y una base de datos básica. En la segunda fase de los ajustes finos, el modelo de detección fue entrenado aún más usando una base de datos específica. Además, la base de datos específica puede ser reemplazada de acuerdo con diferentes aplicaciones. Los experimentos mostraron un desempeño mejorado de la estrategia propuesta en comparación con los ajustes finos realizados tradicionalmente. Una nueva tecnología se ha utilizado recientemente, es llamada aprendizaje débilmente supervisado y se propuso para combatir el problema de escasez de datos, esto abre nuevas investigaciones en el campo de análisis inspección de imágenes.

3.10. El mejoramiento de aprendizaje profundo basado en el dominio del conocimiento de inspección de líneas eléctricas

Las tareas de detección y diagnóstico de los componentes de líneas eléctricas tienen algunos contrastes comparados a la tarea común. Estas características únicas pueden ser denotadas como el conocimiento de dominio en la inspección de líneas eléctricas. En años recientes, se han hecho pocos intentos para mejorar el método de aprendizaje profundo basado en este conocimiento de dominio, lo cual lo hace más capaz para el análisis de datos de inspección de líneas eléctricas. Jiang et al. [34] se concentró en el proceso de detección de las fallas de los aisladores. El algoritmo de diagnóstico tradicional de fallas era usualmente un procedimiento de detección de objetos de dos fases, en el cual primero se detectaba el componente y después se detectaba la falla de la región del componente. Algunos autores señalan que el desempeño del procedimiento tradicional depende del efecto de la detección de componentes, por ejemplo, una vez que se detecta un componente faltante, la identificación de la falla no puede ser alcanzada. Por lo tanto, ellos mejoraron el procedimiento y propusieron un método de diagnóstico de fallas basado en el aprendizaje de ensamblaje con una percepción multinivel. Aplicaron SSD para detectar las tapas faltantes en tres imágenes de entradas diferentes: la imagen aérea original, la imagen con múltiples aisladores y la imagen con un solo aislador. Luego, el resultado final se puede filtrar usando un método de ensamblaje mejorado. En el experimento el procedimiento mejorado mostró una precisión más alta

(92.3%) comparado con el procedimiento tradicional (89.1%), esto verificando la efectividad del método propuesto. Similar a [34], Chen et al. [35] discutió también la mejora del procedimiento de diagnóstico de fallas. Un método de detección de fallas de aisladores se propuso basado en *Second-order Fully Convolutional Network* o SO-FCN. Insertaron una operación de filtrado de imágenes al procedimiento de detección tradicional de dos fases. El procedimiento mejorado consiste en tres pasos principales: el *First-order* FCN fue aplicado para obtener el resultado de segmentación inicial de la región del aislador, el filtrado de reconstrucción morfológica fue realizado para eliminar las identificaciones falsas, luego el *Second-order* FCN se utilizó para detectar las tapas faltantes. Recientemente, Sampedro et al. [36] introdujo una estrategia nueva para detección de tapas faltantes, el cual se transfirió el problema de detección de objetos a un problema de segmentación semántica. El aislador fue conducido por dos elementos, incluyendo las tapas y los conectores que están fuertemente interconectados. Los autores segmentaron las tapas y los conectores de una cadena de aisladores, y generaron una imagen de máscara en la que los pixeles pertenecientes a las tapas fueron cambiados a verde y las regiones de los conectores fueron cambiadas a rojo. En esta imagen de máscara, la detección de las tapas faltantes fue transferida a solo detectar la ausencia de la región verde. Una gran cantidad de muestras de fallos pueden ser sintéticamente producidas eliminando aleatoriamente la región verde de la imagen máscara. En el experimento se usaron un total de 2,400 muestras de entrenamiento, las cuales fueron generadas de 160 imágenes originales, adicionalmente a la optimización del esquema de diagnóstico de fallos. Xiang et al. [37] mejoró la red de aprendizaje profundo. Propusieron una red Faster R-CNN para detectar la fuerza externa de daño como vehículos de las líneas eléctricas. De acuerdo con las características de imágenes de vehículos de ingeniería, por ejemplo, el tamaño del objeto, la forma del objeto y el fondo, los autores modificaron la estructura de la Faster R-CNN para la extracción de características y partes de clasificación. En la extracción de características, una red neuronal convolucional más superficial fue utilizada para extraer características de alta resolución. En la clasificación de dichas características, una capa convolucional fue agregada después de la capa de *pooling* de la *Region of Interest* (RoI) o Región de Interés, para que se pudieran aprender las características a nivel de región que fueran capaces para la RoI. Estas mejoras hicieron que la habilidad de detección de la red se viera beneficiada y las ventajas del método propuesto (89.93%) fue verificado al compararse con la Faster R-CNN tradicional (89.12%).

Capítulo 4

Metodología de Solución

4.1. Generación de la base de datos

El proyecto contempló la utilización de imágenes sintéticas y reales, las primeras habiendo sido obtenidas mediante el uso del motor gráfico Unreal Engine [38] en el que se generaron entornos virtuales, las segundas se obtuvieron de una búsqueda de imágenes con licencia 0 o libre, para evitar alguna infracción de derechos de autor. La base de datos está compuesta de 20 imágenes por clase y se consideraron 3 clases diferentes, resultando en 60 imágenes en total. Cada clase se divide en 2 conjuntos, el de entrenamiento y el de prueba o test, cada uno de estos constituidos de 10 imágenes cada uno. Se utilizarán solamente 5 imágenes para entrenar la red pero para poder obtener estadísticas adicionales como la pérdida por época. Las imágenes del conjunto de prueba son utilizadas en su totalidad para probar diferentes instancias de aisladores. En este caso se requirió crear máscaras adicionales para las imágenes de prueba para calcular el IoU de cada imagen y el mIoU o IoU promedio para cada época.

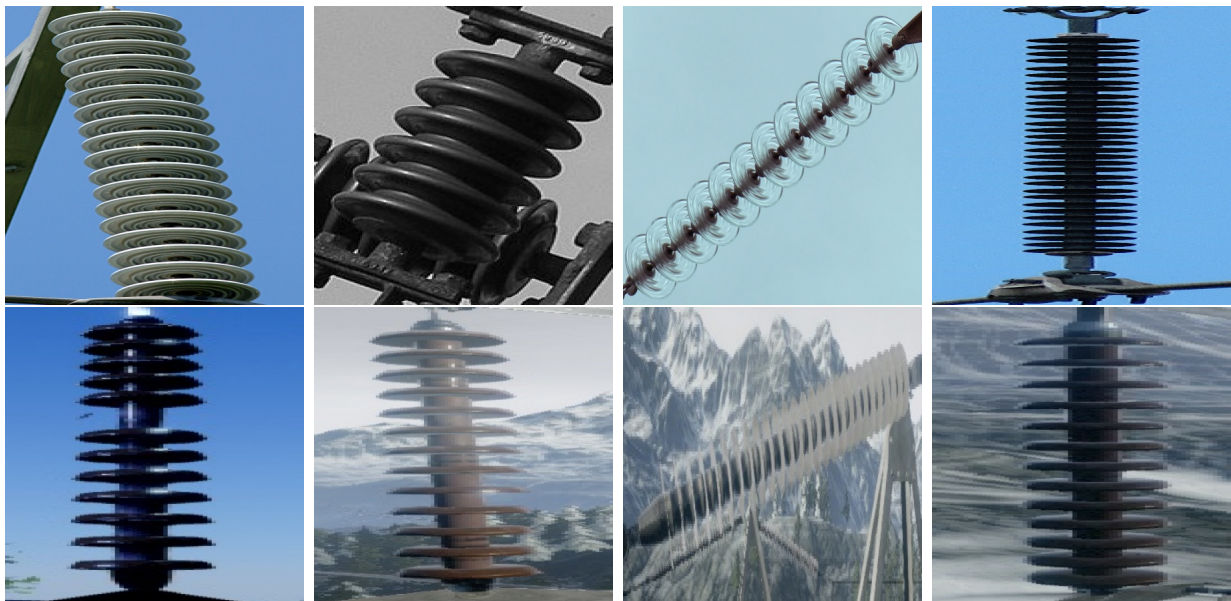


Figura 12. Ejemplos de imágenes sintéticas y reales utilizadas.

4.2. Pre procesamiento de imágenes

Dado que las imágenes obtenidas no se podían utilizar directamente como entradas de la red neuronal por el mismo diseño de esta que solo recibe imágenes cuadradas, fue necesario ajustar el tamaño de las imágenes. También era necesario crear máscaras o segmentación manual para darle una entrada a la red neuronal y pudiera aprender con esas imágenes de soporte ya segmentadas.

4.2.1. Segmentación manual

La segmentación de imágenes manual se hizo para generar las máscaras de segmentación (*ground truth*) que usó la red neuronal para el entrenamiento. Esta consistió en generar con un programa de anotación de imágenes llamado *Image Annotation Tool* [40], esta salida arrojó una imagen en rojo y negro del mismo tamaño que la imagen que se utilizó de entrada, en la cual el rojo indicaba que en ese pixel se encuentra el objeto de interés y el negro señalaba el fondo de la imagen. Seguido de esto se utilizó un programa llamado ImageJ [39] el cual es un procesador de imágenes, en este programa se convirtieron de 8 bits a RGB de 32 bits. Posteriormente a estas imágenes se les cambió el color rojo por blanco para que se tuviera un rango de color más amplio y diferenciar el negro del blanco más fácilmente dado que se utilizó un rango de 0 a 255 para codificar el color, siendo 0 el negro y 255 el blanco. El resultado de este procesamiento son las máscaras que se utilizarán para el entrenamiento.

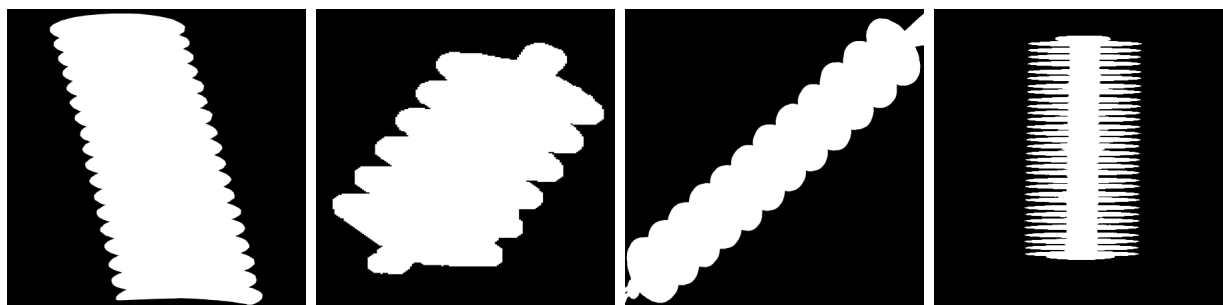


Figura 13. Ejemplos de máscaras generadas.

4.2.2. Recortado y ajuste de tamaño

Después de obtener las máscaras se deben de recortar de tal manera que solo quede el objeto de interés con un poco de fondo en las orillas de este. Para esto se utilizó un programa de creación de cuadros delimitadores llamado *Labelme* [40], este programa creaba a la salida archivos .xml para cada imagen en el que contenía la información de cada cuadro delimitante, así como otros datos de la imagen. Obtenidos estos archivos .xml se creó un código en Python para que, utilizando esa información, generara una imagen nueva por cada cuadro delimitante. Este mismo código recortaba

tanto las imágenes originales como sus máscaras. Después de este proceso se ajustó el tamaño de las imágenes y sus máscaras a un tamaño de 448 por 448 píxeles para poder ser utilizadas como entrada de la red neuronal. Esto se logró utilizando un código propio escrito en Python.

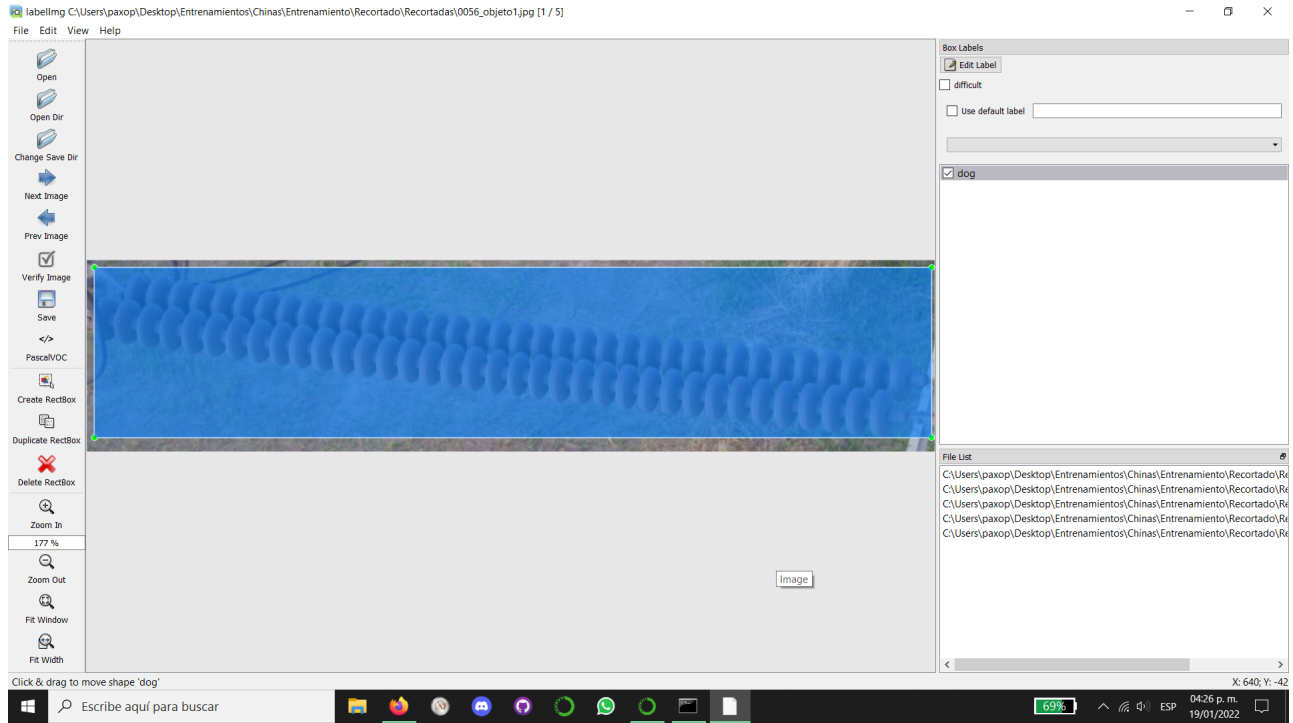


Figura 14. Ejemplo del uso de LabelMe [40]

4.3. Entrenamiento

Se realizaron diferentes entrenamientos para la obtención de mapas de características y redes de relación considerando diferentes tipos de imágenes de entrada. Estos diferentes tipos son: imágenes reales, imágenes sintéticas e imágenes combinadas. Estas últimas son, simplemente, imágenes reales y sintéticas. Los diferentes archivos .pkl de los diferentes entrenamientos nos servirán para observar y comparar los resultados de las diferentes inferencias o segmentaciones para percibir posibles patrones al encontrar problemas o datos adicionales que nos ayuden a determinar la razón por la cual un resultado es diferente a otro, sea mejor o peor. Además de observar cómo se comporta la red con la inclusión de imágenes sintéticas parcial o totalmente con el objetivo de saber si es posible realizar buenas segmentaciones con entrenamiento de imágenes sintéticas para, posiblemente no hacer necesaria la búsqueda de imágenes reales para conseguir una buena segmentación, facilitando el aumento de datos o *data augmentation* que resulta necesario en casos con pocos datos de entrenamiento.

Dado que se busca diseñar un sistema de segmentación utilizando redes neuronales convolucionales con arquitectura *encoder-decoder*, se usó el modelo *few-shot segmentation* propuesto por [42], esto se eligió dada la limitación de la poca cantidad de imágenes que se pueden obtener para resolver el problema. Ya que no todas las imágenes recolectadas pueden ser utilizadas, debido a que esté

mal enfocado el objeto de interés después de una depuración de dichas imágenes se debe de tomar en cuenta la posibilidad de que solo unas cuantas sean útiles. Este modelo muestra que los autores de este artículo fueron capaces de obtener una precisión promedio de 80.12% al evaluar y entrenar con la base de datos FSS-1000 en el caso en el que solamente se utilizan 5 imágenes por clase para entrenamiento, este tipo de entrenamiento también es conocido como 5-shot. Comparando ese resultado con otras redes de segmentación de tipo *few-shot* como la *Guided Network-5shot* y la *OSLSM-5shot* que obtuvieron una precisión promedio de 74.27% y 73.02% respectivamente al ser evaluadas y entrenadas con la misma base de datos FSS-1000. Esto nos da una idea de que, a pesar de la baja precisión comparada con otros métodos de segmentación en los que se utilizan muchas más imágenes para el entrenamiento, si se compara con redes de segmentación con el objetivo de utilizar pocas imágenes para el entrenamiento se puede observar que tiene una mejora considerable en cuanto a la precisión promedio. La arquitectura de la red neuronal convolucional utilizada se compone de 3 partes, un módulo codificador, un módulo decodificador y un módulo de relación como se puede observar en la figura 15. En el módulo de relación se puede dividir en dos partes, la primera en la que los mapas de características generados por el módulo codificador, tanto del conjunto de consulta como del de entrenamiento, se combinan mediante la operación de concatenación para lograr que se considere el doble de mapas de características, logrando así una mejor puntuación de relación, la cual se realizará en la siguiente parte. En la segunda parte, se genera una puntuación de relación en un rango de 0 a 1, el cual representa la similitud entre cada imagen de consulta con cada imagen de entrenamiento. Esto resulta en 5 puntuaciones de relación para cada imagen de consulta, indicando qué tan parecida es esta en comparación con cada una de las imágenes de entrenamiento.

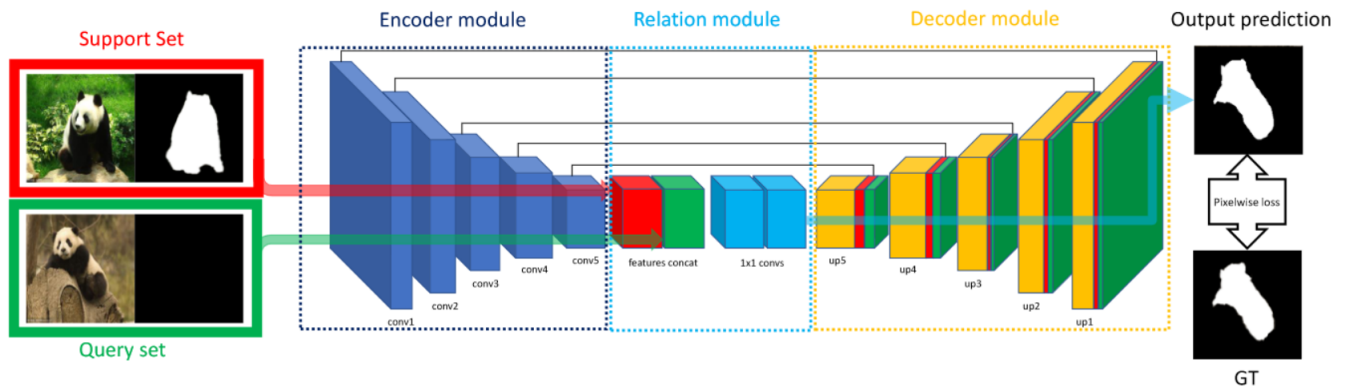


Figura 15. Arquitectura *encoder-decoder* utilizada. Obtenida de [42]

Se creará una base de datos con imágenes aéreas de aisladores en torres eléctricas, así como de forma sintética al ser tomadas utilizando un entorno virtual a través del motor gráfico *Unreal Engine 4* [38] para observar cómo se comporta la red neuronal en esas situaciones, experimentando sobre qué tan precisa es la segmentación de las imágenes virtuales al entrenar el modelo con imágenes de aisladores reales y viceversa. Estos resultados pueden ser de importancia para futuras obtenciones de bases de datos de dichos aisladores, dado que en ciertas situaciones es más fácil obtener una amplia variedad de imágenes utilizando un entorno virtual, además del menor costo que supone comparado con el uso de drones aéreos u otros métodos que se utilizan para la captura de las

imágenes aéreas como drones escaladores o el uso de helicópteros.

Se tomará un número determinado de imágenes para recortarlas y segmentarlas manualmente, obteniendo así el *ground-truth* que se necesita para el entrenamiento *few-shot* que se utilizará. Para esto se obtendrán dichas imágenes de bases de datos que permitan el uso de estas, sin infringir en los derechos de autor, es decir, utilizando solamente imágenes con licencia cero o de dominio público. En el caso de las imágenes en el entorno virtual utilizando el motor gráfico *Unreal Engine* ya que son imágenes y modelos tridimensionales tales que se permite el uso de estos al público en general con tan solo pagar por dichos modelos y entorno.

Posteriormente a la creación de la base de datos, se comenzará a tratar las imágenes de tal manera que se puedan utilizar para el entrenamiento. Para esto, se utilizará un programa de anotación de imágenes en conjunto con un programa de segmentación manual de imágenes, esto además de un código propio de recorte de objetos en las imágenes que toma como parámetros las salidas un programa de creación de archivos tipo XML con la información de varias áreas determinadas de interés [40]. Todo esto resultará en dos conjuntos de archivos que se proporcionarán a la red neuronal para el entrenamiento, así como para la inferencia que se hará después. Estos conjuntos de archivos serán las imágenes que se quieren utilizar para entrenar y probar la red neuronal junto con sus respectivos *ground-truth*.

Primeramente, se tomará un cierto número de imágenes y se recortarán con los programas antes mencionados, el programa que genera archivos XML con una herramienta de segmentación de imágenes, para que el programa que recorte y genere nuevos archivos PNG de las segmentaciones indicadas las suministre al programa FIJI [39], el cual procesa las imágenes de tal manera que las podremos utilizar como capas de segmentación para el *ground-truth*, el cual necesitaremos para el último programa que detecta dichas capas como máscaras de segmentación y FIJI nos ayudará a facilitar el proceso de la creación de máscaras dado que solo tenemos que corregir ciertas imperfecciones con el último programa.

Con el programa *Image Annotation Tool* se realizarán las correcciones necesarias a las máscaras de las imágenes que se utilizarán para el entrenamiento en la red tipo *encoder-decoder*, este programa nos permite marcar píxel a píxel la máscara tanto para borrar partes de ella que están en donde no les corresponde o colocar nuevos pixeles a la máscara en partes que queremos que la red tome como referencia para el entrenamiento.

La red tomará aleatoriamente un número determinado de imágenes de la carpeta en donde se encuentran las imágenes con sus respectivas máscaras de *ground-truth*, es decir, máscaras que fueron etiquetadas manualmente. Con estas imágenes se llevará a cabo el proceso de entrenamiento y por último se utilizará una de las muchas salidas de la red ya que ésta tiene un parámetro para controlar cada cuántas épocas crea una salida (cada salida se compone de una *relation network* y un *feature encoder*) y cuántas épocas realizará en total, esto ayudará a realizar la predicción de imágenes que contienen el objeto de interés, en este caso aisladores eléctricos, para observar y evaluar la precisión de dicha predicción y entonces determinar si se utilizará otra salida del mismo entrenamiento para realizar dichas predicciones. El modelo se compone de 3 módulos y se basa en la arquitectura VGG16 para la extracción de características dado que la última capa de mapas de características contenía suficientes características del objeto de interés, las cuales pueden ser decodificadas por medio de una red de *upsampling* para producir la segmentación resultante. El módulo codificador realiza convoluciones de tal manera que, dada una entrada, genera mapas de características para el conjunto de soporte, así como para el conjunto de prueba. Luego se combinan dichos mapas de características en el módulo de relación, esto se logra utilizando la operación de

concatenación. Este módulo genera una segmentación poco precisa en baja resolución basado en los mapas de características. Finalmente, este resultado poco preciso se toma como entrada para el módulo de decodificador para recuperar el mapa de predicción a la misma resolución de la entrada del módulo codificador por medio de bloques de *upsampling* que consisten en la utilización de una capa de *nearest neighbor* junto con una capa de convolución.

4.4. Métricas de evaluación

La evaluación de la precisión de dichas predicciones se realizará utilizando la métrica intersección sobre unión o IoU (por sus siglas en inglés *Intersection over Union*)

El IoU está definido como un valor entre 0 y 1 que especifica la cantidad de solapamiento entre la predicción y el *ground truth*, cuanto más cercano sea el valor a 1, mayor es el solapamiento. Esta medida de precisión es de gran importancia cuando se utiliza anotación manual humana como *ground truth*. [46]


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figura 16. Fórmula para la obtención del IoU. Obtenida de [45]

Capítulo 5

Resultados

Los resultados que se obtuvieron después de los varios entrenamientos y diferentes inferencias fueron las siguientes y se mostrarán mediante las cuatro mejores y cuatro peores imágenes resultantes de cada entrenamiento con cada inferencia o segmentación automática. Algunos ejemplos de los resultados considerando diferentes tipos de entrenamiento se pueden observar en la figura 17.

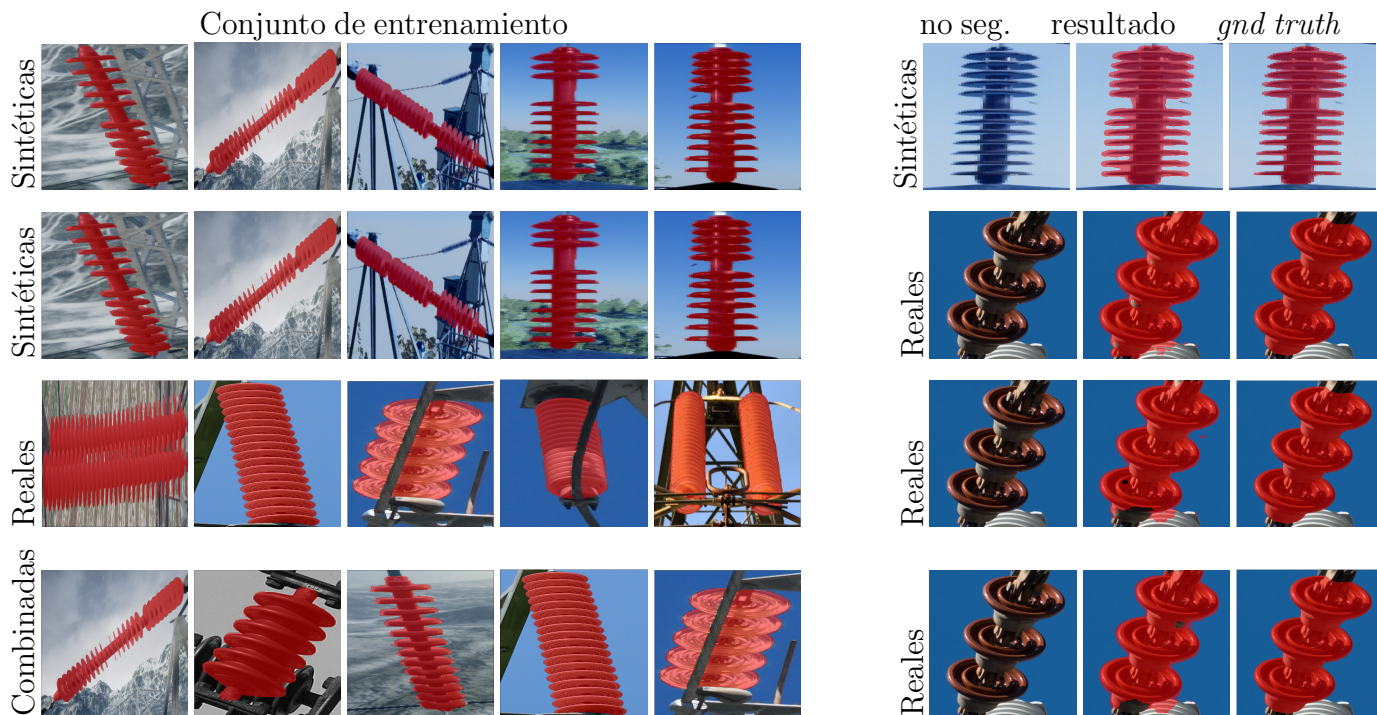


Figura 17. A la izquierda se observan las imágenes utilizadas para el entrenamiento y a la derecha la imagen sin segmentar, el resultado y el *ground truth*

5.1. Caso 1: sintéticas - sintéticas

En este caso se entrenó la red solamente con imágenes sintéticas y se hizo la inferencia en imágenes sintéticas. Esto se realizó para lograr observar el desempeño que tenía la red en estas condiciones relativamente controladas al ser imágenes obtenidas de un entorno virtual tanto el conjunto de entrenamiento como el de prueba. Este primer experimento arrojó imágenes con buena segmentación así como otras no tan buenas. Se encontraron algunas posibles situaciones en las que los resultados no se obtenían tan bien como objetos metálicos como partes de una torre eléctrica o mucho ruido visual en el fondo como árboles. En general se obtuvieron buenos resultados individuales y aunque el promedio general no es tan bueno por algunos valores de IoU muy bajos, es un avance positivo, ya que se están considerando solamente 5 imágenes para el entrenamiento. El desempeño del IoU se puede observar en la figura 18.

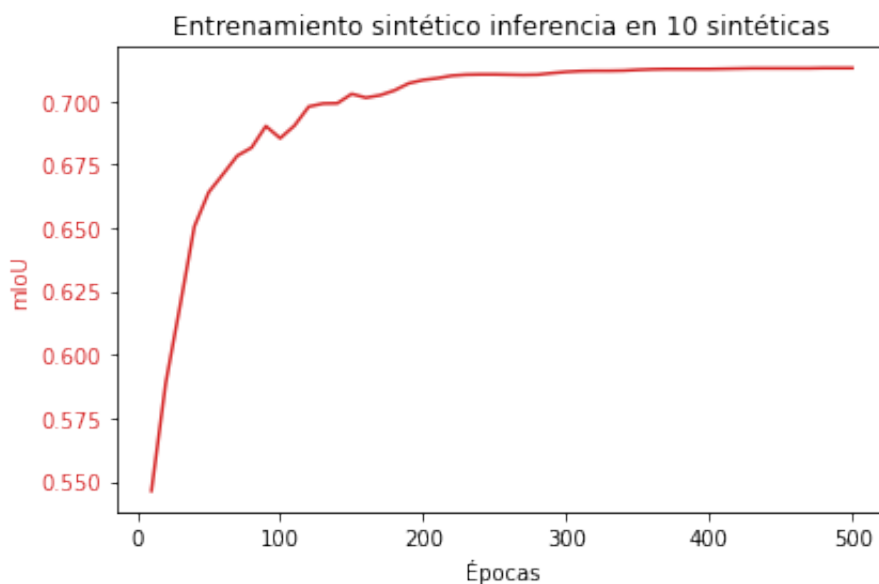
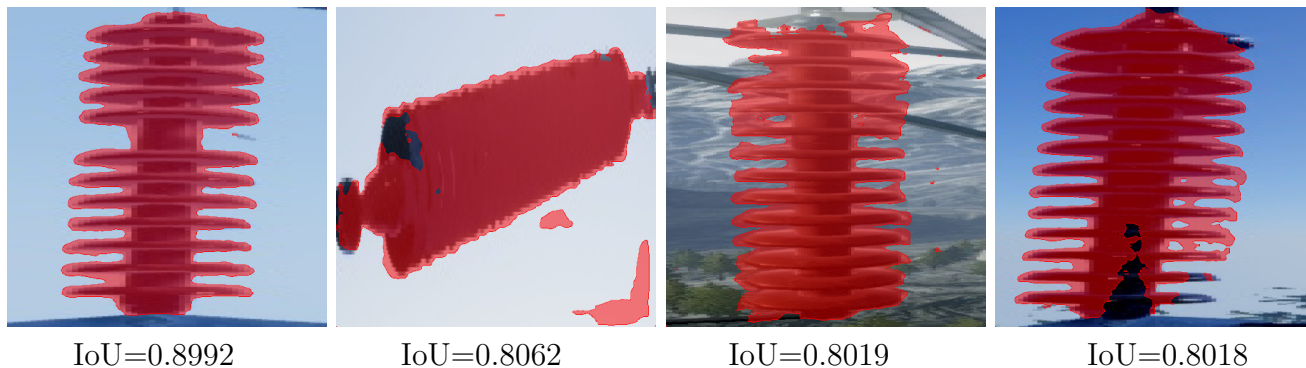


Figura 18. Comportamiento del mIoU en el conjunto de test para el caso 1: sintéticas-sintéticas



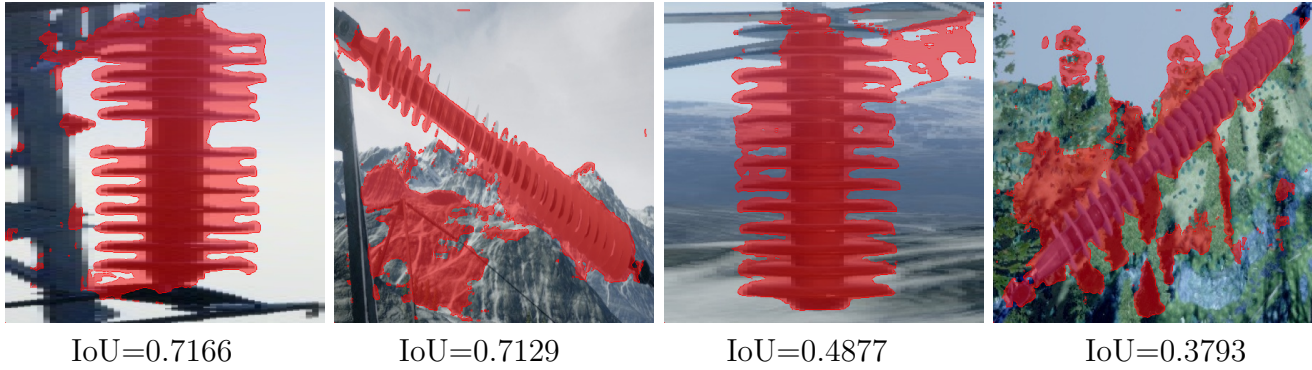


Figura 19. Resultados obtenidos del caso 1. La primera fila corresponde a las imágenes con los 4 valores más altos de IoU y la segunda a las imágenes con los 4 valores más bajos de IoU.

5.2. Caso 2: sintéticas - reales

Al obtener buenos resultados con las imágenes sintéticas se intentó ahora realizar la inferencia con el conjunto de imágenes reales. Se obtuvieron resultados peores al entrenamiento anterior aunque eran esperados dado que el conjunto de entrenamiento no tomaba en cuenta imágenes reales. Pero se mantuvieron buenos resultados en algunas imágenes a pesar de tener un par de resultados por debajo del 50 %. Se notan también los mismos posibles causantes de las imágenes con baja precisión al realizarse la segmentación. El IoU promedio por época se muestra en la figura 20.

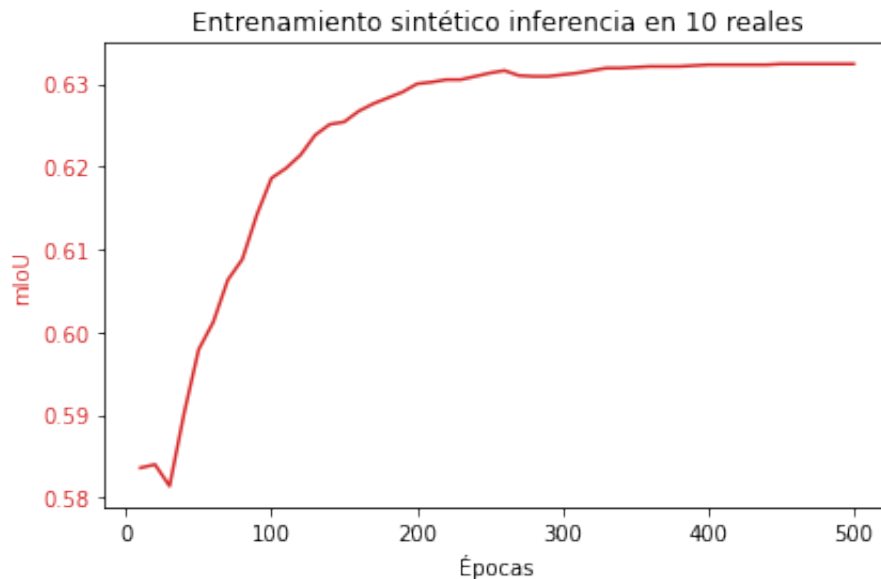


Figura 20. Comportamiento del mIoU en el conjunto de test para el caso 2: sintéticas-reales.

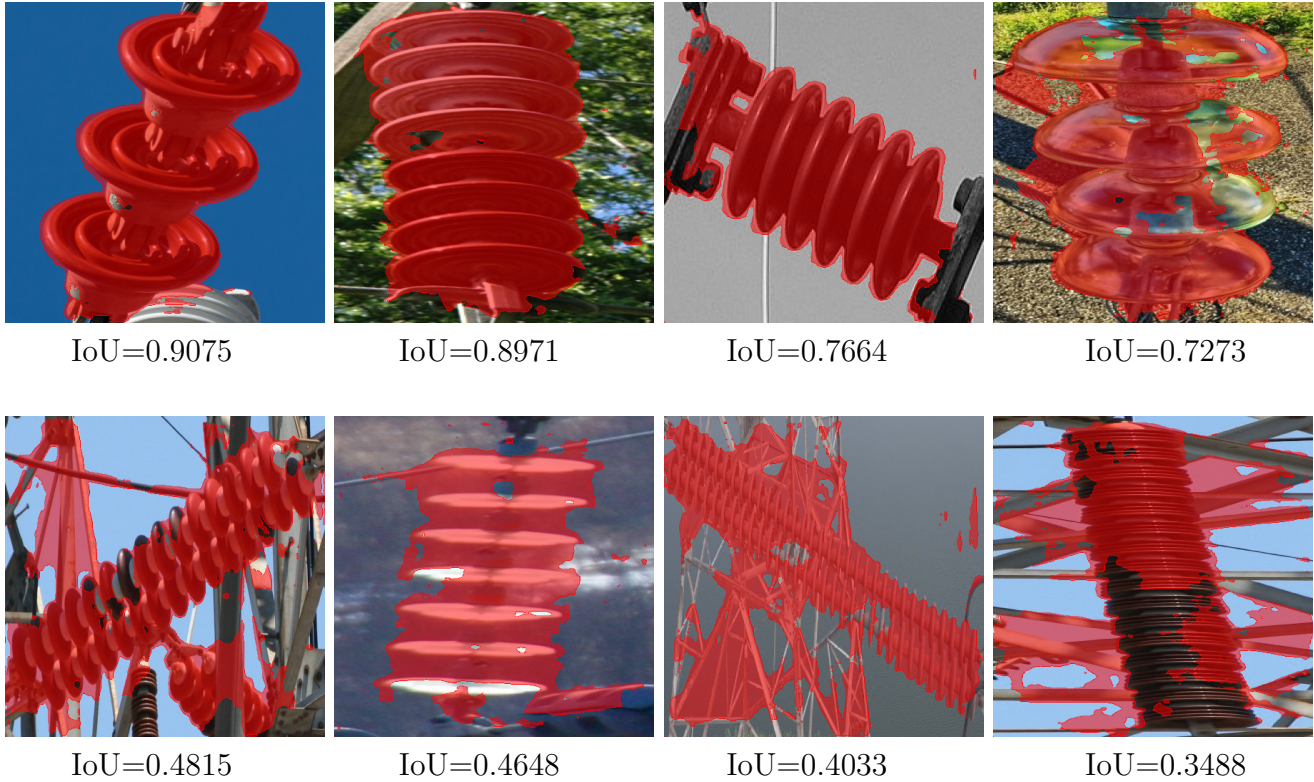


Figura 21. Resultados obtenidos del caso 2. La primera fila corresponde a las imágenes con los 4 valores más altos de IoU y la segunda a las imágenes con los 4 valores más bajos de IoU.

5.3. Caso 3: reales - reales

En este caso se buscó un caso más real, es decir, lo más sencillo de hacer para un quien quiere segmentar imágenes de aisladores eléctricos en una subestación eléctrica sería quizá utilizar un dron para tomar fotos a los aisladores y con ellas formar tanto el conjunto de entrenamiento como el de prueba. A pesar de que se pensó que habría un desempeño mejor en comparación con el entrenamiento anterior hecho con imágenes sintéticas se observó lo contrario. Es posible que el hecho de que los aisladores reales pueden ser fabricados con diferentes materiales, dificulta más su segmentación. El comportamiento del IoU a través de las épocas de entrenamiento se puede observar en la figura 22.

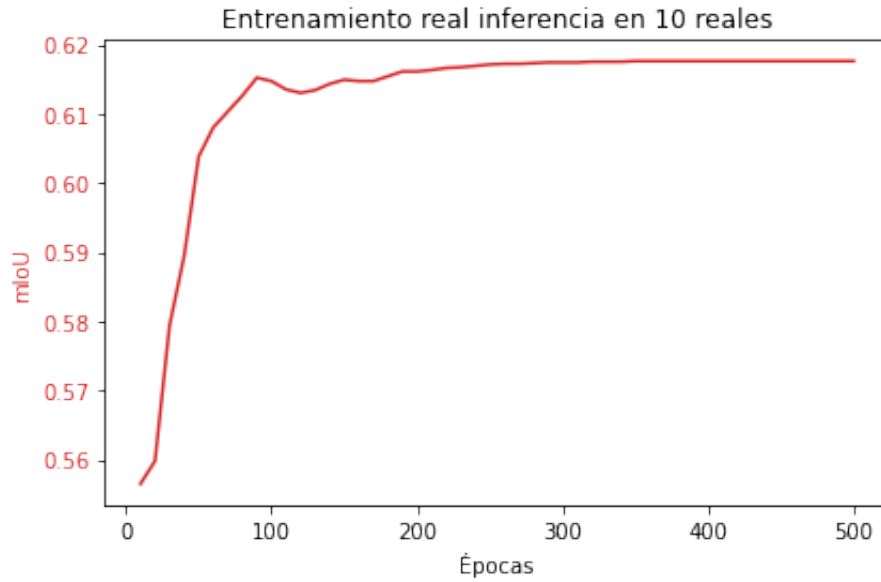


Figura 22. Comportamiento del mIoU en el conjunto de test para el caso 3: reales-reales.

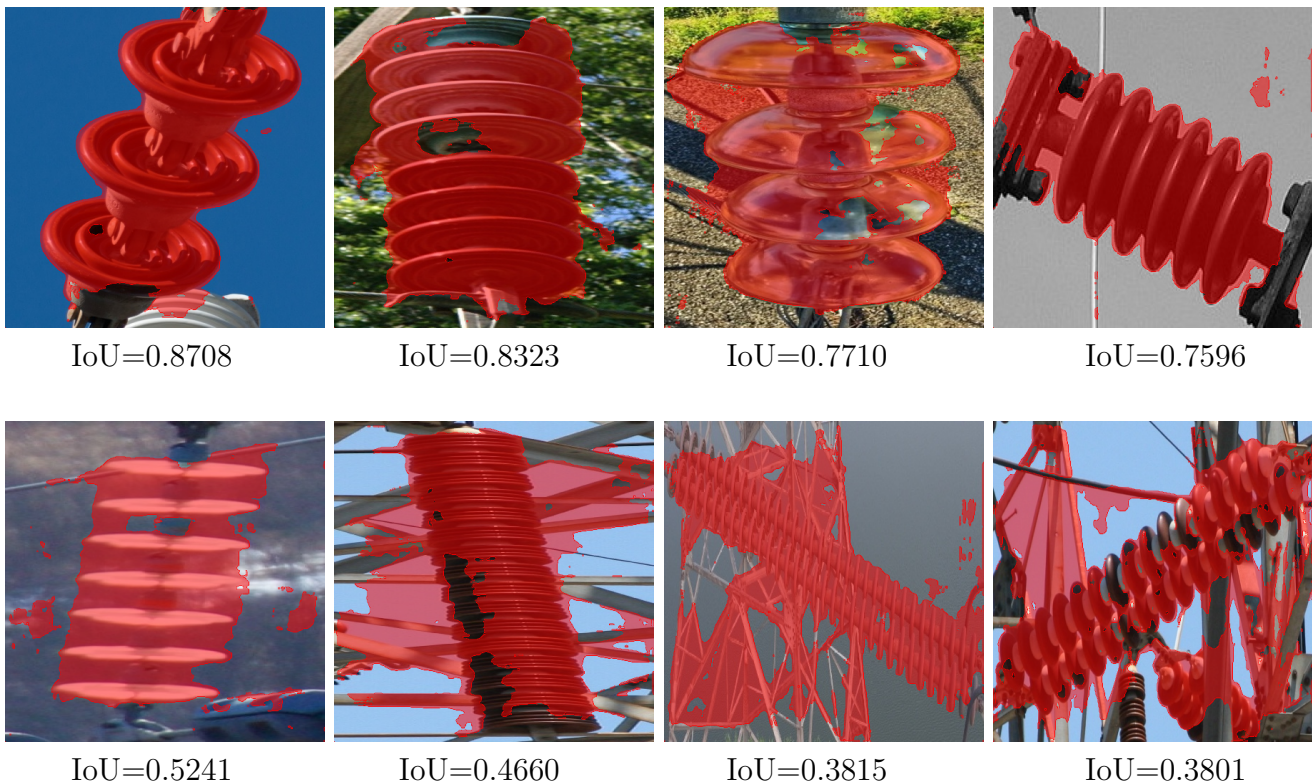


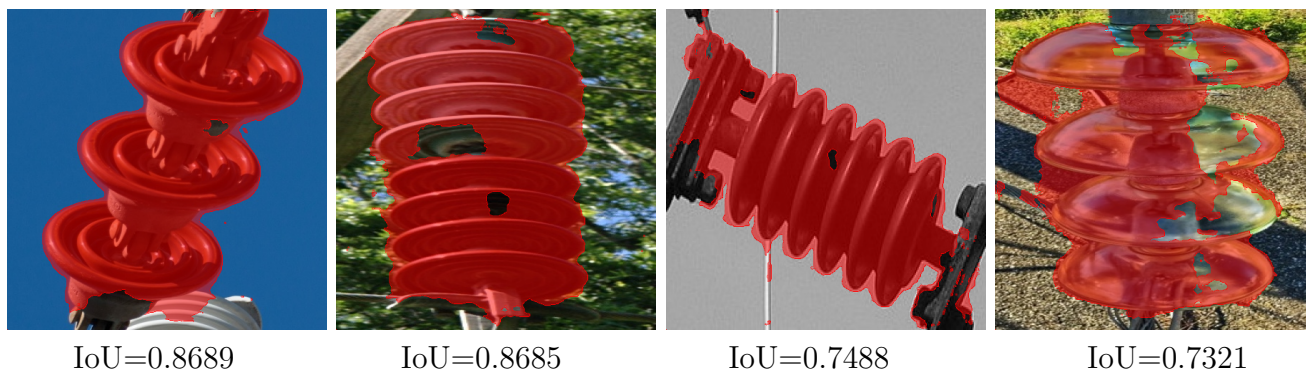
Figura 23. Resultados obtenidos del caso 3. La primera fila corresponde a las imágenes con los 4 valores más altos de IoU y la segunda a las imágenes con los 4 valores más bajos de IoU.

5.4. Caso 4: combinadas - reales

Para este caso se pensaba que mejorarían los resultados del caso anterior, ya que se utilizaron imágenes sintéticas, las cuales tuvieron el mejor desempeño al realizar la segmentación sobre imágenes reales. Cabe mencionar que las imágenes reales en el conjunto de prueba han sido las mismas tanto para el caso del entrenamiento con imágenes reales como con imágenes sintéticas. Esto para poder tener una mejor comparación entre esos dos entrenamientos, así como de este último entrenado con imágenes reales y sintéticas. Pero se obtuvieron resultados aún peores que el entrenamiento solamente con imágenes reales. Es posible que las imágenes sintéticas utilizadas en conjunción con las reales no fueron suficientes para que la red considerara los distintos materiales de los que están fabricados los aisladores reales, a esto se le puede atribuir la disminución en la precisión, adicionalmente, se pudieron observar otros posibles factores que hemos visto en todos los casos anteriores que causan una menor precisión al estar presentes. Los resultados de precisión del IoU general se muestra en la figura 24.



Figura 24. Comportamiento del mIoU en el conjunto de test para el caso 4: combinadas-reales.



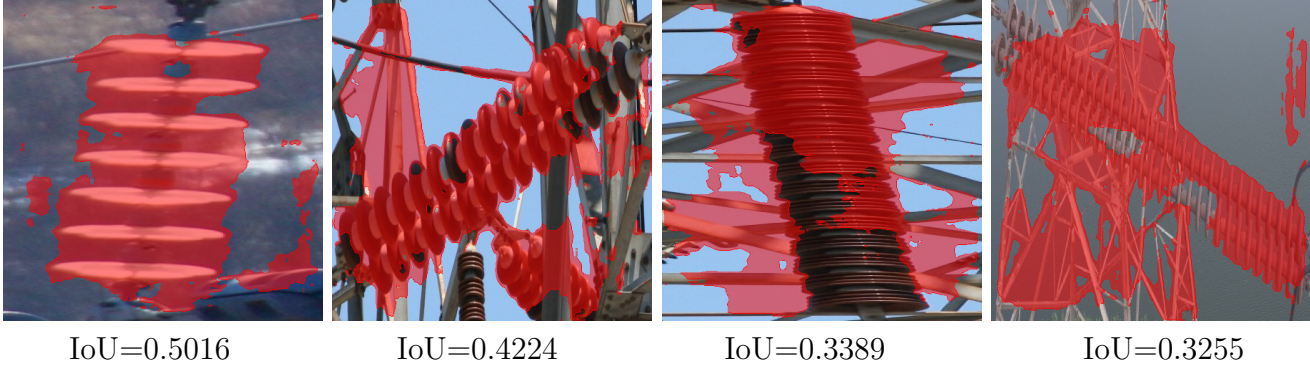


Figura 25. Resultados obtenidos del caso 4. La primera fila corresponde a las imágenes con los 4 valores más altos de IoU y la segunda a las imágenes con los 4 valores más bajos de IoU.

5.5. Análisis de todos los resultados obtenidos

Al final podemos concluir que los experimentos fueron exitosos, se puede conseguir una buena segmentación al usar imágenes sintéticas para el entrenamiento. Otros detalles que pudimos observar son que la segmentación mejora sustancialmente si se usan en el conjunto de prueba imágenes que estén sobre un fondo uniforme, es decir, que el objeto de interés se diferencie mucho del fondo o que el fondo no sea tan variado o con tantos detalles como árboles, cambios de texturas o colores. Aunque esta última condición en las fotos reales de un dron no pueden ser siempre satisfechas, sí se tiene una referencia de qué tipo de fotos se buscan obtener si es que se busca utilizar un dron aéreo. Aunque no se realiza una segmentación tan buena comparada con los métodos actuales que priorizan precisión, nuestro modelo que prioriza eficiencia en el uso de las imágenes de entrenamiento hace un buen trabajo que oscila entre los 60 y 70 por ciento en cuestión de precisión, lo cual nos indica que es posible que otro tipo de red pueda mejorar dichos resultados o que aumentando el tamaño de las imágenes de entrada post ajuste de tamaño, la precisión aumente, también es importante notar que aumentando relativamente poco el número de imágenes con las que se entrena la red, que en este caso fue *few-shot learning*, específicamente *5-shot learning*, por ejemplo, es muy posible que utilizando 10 o 20 imágenes para entrenamiento la misma red tenga mejores resultados en cuanto a precisión, esto debido a que existen diferentes modelos de aisladores. Algunos otros enfoques en trabajos posteriores podrían ser solo entrenar la red con imágenes de un solo tipo de aislador y no con una gran variedad de aisladores en términos de materiales y formas, así como también entrenar con imágenes que tengan mucha más variedad de entornos para que los problemas de ruido visual pudieran potencialmente causar menos disminución en la precisión. El resumen de los valores del IoU general para cada caso se pueden observar en la tabla 1.

Tabla 1. Porcentaje de mIoU por tipo de entrenamiento e inferencia.

Entrenamiento	Inferencia	mIoU (%)
Sintéticas	Sintéticas	71.31
Sintéticas	Reales	63.24
Reales	Reales	61.76
Combinadas	Reales	60.13

Capítulo 6

Conclusiones

Después de analizar todos los resultados y observar los patrones que proporcionan mejores segmentaciones, así como los que empeoran dichas segmentaciones, se puede concluir que el modelo utilizado puede ser una buena opción para realizar una buena segmentación de los aisladores eléctricos. El hecho de que se hayan utilizado pocas imágenes **los resultados obtenidos son comparables con los generados mediante métodos de segmentación del estado del arte en otros conjuntos de datos. En casos en los que no se cuente con tantos datos o imágenes para el entrenamiento, esta red puede ser una buena manera de aproximarse a el problema de segmentación. Adicionalmente, se encontraron posibles referencias a seguir para la obtención de las imágenes que se desean segmentar, aunque no siempre se pueden conseguir las condiciones previamente mencionadas, es posible mejorar los resultados siguiendo estas referencias.

Con respecto a los objetivos específicos, la obtención de imágenes sintéticas mediante el motor gráfico *Unreal Engine 4* fue exitosa y relativamente sencilla, aunque el entorno virtual que se utilizó en esta tesis no fue gratuito. También se pudo observar que estas imágenes tuvieron los mejores resultados tomando en cuenta el valor IoU general, esto podría significar que se pueden utilizar para entrenamiento y segmentar imágenes reales con mejor precisión, lo cual es bueno porque este tipo de imágenes se pueden manipular con más facilidad que las reales y se pueden seguir las referencias desde la creación de los entornos virtuales dentro del *Unreal Engine 4*, haciendo la obtención de las imágenes sintéticas mucho más óptima dado que se puede escoger el ángulo de la captura de dicha imagen, se puede manipular el fondo para que esté limpio y procurando que haya el mínimo de ruido visual, haciendo de estas imágenes mejores para la fase de entrenamiento.

Capítulo 7

Referencias

1. Deng, L., Yu, D. (2014), "Deep Learning: Methods and Applications", Foundations and Trends in Signal Processing: Vol. 7: No. 3–4, 197–387. Now publishers. doi:10.1561/20000000039
2. Bengio, Y. (2009). "Learning Deep Architectures for AI". Foundations and Trends in Machine Learning. Vol. 2 No. (1), 1–127. Now publishers. doi: 10.1561/22000000006
3. Bengio, Y., Courville, A., Vincent, P. (2013). Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 35 No. (8): 1798–1828. doi: 10.1109/TPAMI.2013.50
4. LeCun, Y., Bengio, Y., Hinton, G. (2015). "Deep learning". Nature. Vol. 521 No. (7553): 436–444
5. Badrinarayanan, V., Kendall, A., Cipolla, R. (2017). "Segnet: a deep convolutional encoder-decoder architecture for image segmentation". IEEE Transactions on Pattern Analysis and Machine Intelligence. 39 (12), 2481–2495
6. Mayer, N., Ilg, E., Haussler, P., Fischer, P., Cremers, D., Dosovitskiy, A., et al. (2016). "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". 2016 IEEE Conference on Computer Vision and Pattern Recognition 4040–4048. doi: 10.1109/CVPR.2016.438
7. Yang, J., Price, B., Cohen, S., Lee, H., Yang, M. H. (2016). "Object contour detection with a fully convolutional encoder-decoder network". 2016 IEEE Conference on Computer Vision and Pattern Recognition. 193–202. ArXiv. doi: 10.1109/CVPR.2016.28
8. Adhikari, S.P., Yang, H., Kim, H. (2019) "Learning Semantic Graphics Using Convolutional Encoder–Decoder Network for Autonomous Weeding in Paddy". Front. Plant Sci. 10:1404. doi: 10.3389/fpls.2019.01404
9. Shapiro, L., Stockman, G. (2001): "Computer Vision", pp 279–325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

-
10. Barghout, L., Lee, L. (2004) "Perceptual information processing system." Paravue Inc. U.S. 10.618543/abandoned.
 11. Zachow, S., Zilske, M., Hege, H.C. (2007). '3D reconstruction of individual anatomy from medical image data: Segmentation and geometry processing' ANSYS Conference & 25. CAD-FEM Users Meeting November 2007, Internationales Congress Center Dresden Converge. Zuse Institute Berlin. urn: urn:nbn:de:0297-zib-10440.
 12. Dazhou, G., Yanting, P., Kang, Z., Hongkai, Y., Yuhang, L., Song, W. (2019). "Degraded Image Semantic Segmentation With Dense-Gram Networks". IEEE Transactions on Image Processing. 29: 782–795. doi: 10.1109/TIP.2019.2936111
 13. Jingru, Y., Pengxiang, W., Menglin, J., Qiaoying, H., Hoepfner, D. J., Metaxas, D. N. (2019). "Attentive neural cell instance segmentation". Medical Image Analysis. 55: 228–240. doi: 10.1016/j.media.2019.05.004.
 14. Liu, X., Jiang, H., Chen, J., Chen, J., Zhuang, S., Miao, X. (2018) "Insulator detection in aerial images based on faster regions with convolutional neural network,". IEEE 14th International Conference on Control and Automation (ICCA). 1082–1086. doi: 10.1109/ICCA.2018.8444172
 15. Liu, X., Jiang, H., Chen, J., Miao, X. (2020) "Review of data analysis in vision inspection of power lines with an in-depth discussion of deep learning technology" in 2020. Annual Reviews in Control Volume 50, 2020, 253-277. doi: 10.1016/j.arcontrol.2020.09.002
 16. Fei-Fei, L., Krishna, R., Danfei X. (2021). "cs231n", GitHub Repository. <https://cs231n.github.io/convolutional-networks>.
 17. Bian, J., Hui, X., Zhao, X., Tan, M. (2019). "A monocular vision-based perception approach for unmanned aerial vehicle close proximity transmission tower inspection," International Journal of Advanced Robotic Systems, vol. 16, no. 1. doi: <https://doi.org/10.1177/1729881418820227>
 18. Hui, X., Bian, J., Zhao, X., Tan, M. (2018). "Vision-based autonomous navigation approach for unmanned aerial vehicle transmission-line inspection," International Journal of Advanced Robotic Systems, vol. 15, no. 1, doi: <https://doi.org/10.1177/1729881417752821>
 19. Wang, W., Tian, B., Liu, Y., Liu, L., Li, J. (2017). "Study on the electrical devices detection in uav images based on region based convolutional neural networks," Journal of Geo-information Science, vol. 19, no. 2, pp. 256–263. doi: 10.3724/SP.J.1047.2017.00256
 20. Xu, C., Bo, B., Liu, Y., Tao, F. (2018). "Detection method of insulator based on single shot multibox detector," in Journal of Physics: Conference Series, vol. 1069, no. 1. IOP Publishing, p. 012183. doi: <https://doi.org/10.1088/1742-6596/1069/1/012183>
 21. Wang, S., Niu, L., Li, N. (2018). "Research on image recognition of insulators based on yolo algorithm," in 2018 International Conference on Power System Technology (POWERCON). IEEE, pp. 3871–3874. doi: 10.1109/POWERCON.2018.8602149

-
22. Chen, B., Miao, X. (2019). "Distribution line pole detection and counting based on yolo using uav inspection line video," *Journal of Electrical Engineering & Technology*, 15, no. 3, pp. 1–8. doi: 10.1007/s42835-019-00230-w
 23. Chang, W., Yang, G., Li, E., Liang, Z. (2018). "Toward a cluttered environment for learning-based multi-scale overhead ground wire recognition," *Neural Processing Letters*, vol. 48, no. 3, pp. 1789–1800. doi: <https://doi.org/10.1007/s11063-018-9799-3>
 24. Chang, W., Yang, G., Yu, J., Liang, Z. (2018). "Real-time segmentation of various insulators using generative adversarial networks," *IET Computer Vision*, vol. 12, no. 5, pp. 596–602. doi: 10.1049/iet-cvi.2017.0591
 25. Zhao, Z., Xu, G., Qi, Y., Liu, N., Zhang, T. (2016). "Multi-patch deep features for power line insulator status classification from aerial images," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3187–3194. doi: 10.1109/IJCNN.2016.7727606
 26. Maeda, K., Takahashi, S., Ogawa, T., Haseyama, M. (2018). "Estimation of deterioration levels of transmission towers via deep learning maximizing canonical correlation between heterogeneous features," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 633–644. doi: 10.1109/JSTSP.2018.2849593
 27. Yang, Y., Wang, L., Wang, Y., Mei, X. (2019). "Insulator self-shattering detection: a deep convolutional neural network approach," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 10 097–10 112. doi: <https://doi.org/10.1007/s11042-018-6610-4>
 28. Gao, F., Wang, J., Kong, Z., Wu, J., Feng, N., Wang, S., Hu, P., Li, Z., Huang, H., Li, J. (2017). "Recognition of insulator explosion based on Deep learning," in *2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2017, pp. 79–82. doi: 10.1109/ICCWAMTIP.2017.8301453
 29. Ling, Z., Qiu, R. C., Jin, Z., Zhang, Y., He, X., Liu, H., Chu, L. (2018). "An accurate and real-time self-blast glass insulator location method based on faster r-cnn and u-net with aerial images," *arXiv preprint arXiv:1801.05143*, 2018. arXiv:1801.05143
 30. Tao, X., Zhang, D., Wang, Z., Liu, X., Zhang, X., Xu, D. (2018). "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018. doi: 10.1109/TSMC.2018.2871750
 31. Bai, R., Cao, H., Yu, Y., Wang, F., Dang, W., Chu, Z. (2018). "Insulator fault recognition based on spatial pyramid pooling networks with transfer learning (match 2018)," in *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2018, pp. 824–828. doi: 10.1109/ICARM.2018.8610720
 32. Miao, X., Liu, X., Chen, J., Zhuang, S., Fan, J., Jiang, H. (2019). "Insulator detection in aerial images for transmission line inspection using single shot multibox detector," *IEEE Access*, vol. 7, pp. 9945–9956, 2019. doi: 10.1109/ACCESS.2019.2891123

-
33. Lee, S. J., Yun, J. P., Choi, H., Kwon, W., Koo, G., Kim, S. W. (2017). “Weakly supervised learning with convolutional neural networks for power line localization,” in 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2017, pp. 1–8. doi: 10.1109/SSCI.2017.8285410
 34. Jiang, H., Qiu, X., Chen, J., Liu, X., Miao, X., Zhuang, S. (2019). “Insulator fault detection in aerial images based on ensemble learning with multi-level perception,” IEEE Access, vol. 7, pp. 61 797–61 810, 2019. doi: 10.1109/ACCESS.2019.2915985
 35. Chen, J., Xu, X., Dang, H. (2019). “Fault detection of insulators using second-order fully convolutional network model,” Mathematical Problems in Engineering, vol. 2019, 2019. doi: <https://doi.org/10.1155/2019/6397905>
 36. Sampedro, C., Rodriguez-Vazquez, J., Rodriguez-Ramos, A., Carrio, A., Campoy, P. (2019). “Deep learning-based system for automatic recognition and diagnosis of electrical insulator strings,” IEEE Access, vol. 7, pp. 101 283–101 308, 2019. doi: 10.1109/ACCESS.2019.2931144
 37. Xiang, X., Lv, N., Guo, X., Wang, S., El Saddik, A. (2018). “Engineering vehicles detection based on modified faster r-cnn for power grid surveillance,” Sensors, vol. 18, no. 7, p. 2258, 2018. doi: <https://doi.org/10.3390/s18072258>
 38. Unreal Engine 4 (N^o de versión 4.26). (2015). Windows. North Carolina, USA: Epic Games.
 39. Rasband, W. (1997). ImageJ (N^o de versión 1.52p). Windows. Maryland, USA: National Institutes of Health.
 40. Russell, B. C., Torralba, A., Murphy, K. P., Freeman, W. T. (2008). LabelMe Annotation Tool (N^o de versión 4.5.9). Windows. Massachusetts, USA: National Institutes of Health.
 41. Shama S. (2017). “Activation Functions in Neural Networks. In towards data science. Recuperado de: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
 42. Li, X., Wei, T., Chen, Y. P., Tai, Y., Tang, C. (2020). “FSS-1000: A 1000-Class Dataset for Few-Shot Segmentation”. in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/CVPR42600.2020.00294
 43. Yang, L., Fan, J., Liu, Y., Li, E., Peng, J., Liang, Z. (2020). “A Review on State-of-the-Art Power Line Inspection Techniques”. In IEEE Transactions on Instrumentation and Measurement, 69, No. 12, 9350-9365. doi: 10.1109/TIM.2020.3031194
 44. Liu, P. L. (2020) “Single Stage Instance Segmentation – A Review”. In towards data science. Recuperado de: <https://towardsdatascience.com/single-stage-instance-segmentation-a-review-1eeb66e0cc49>
 45. Rosebrock, A. (2016). “Intersection over Union (IoU) for object detection”. In pymagesearch. Recuperado de: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

-
46. Hofesmann, E. (2020). "IoU a better detection evaluation metric". In towards data science. Recuperado de:
<https://towardsdatascience.com/iou-a-better-detection-evaluation-metric-45a511185be1>



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Instituto de
Investigación en
Ciencias
Básicas y
Aplicadas

INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS



Control Escolar de Licenciatura

VOTOS DE APROBATORIOS

Secretaria Ejecutiva del Instituto de Investigación en Ciencias Básicas Aplicadas de la Universidad Autónoma del Estado de Morelos.
P r e s e n t e .

Por medio de la presente le informamos que después de revisar la versión escrita de la tesis que realizó el C. **AQUINO FLORES FRANCISCO** con número de matrícula **10003167** cuyo título es:

"Segmentación automática de aisladores eléctricos usando pocas imágenes para el entrenamiento"

Consideramos que **SI** reúne los méritos que son necesarios para continuar los trámites para obtener el título de **LICENCIADO EN CIENCIAS ÁREA TERMINAL EN CIENCIAS COMPUTACIONALES Y COMPUTACIÓN CIENTÍFICA.**

Cuernavaca, Mor a 09 de mayo de 2022

Atentamente
Por una universidad culta

Se adiciona página con la e-firma UAEM de los siguientes:

Dr. Jorge Hermosillo Valadez
Dra. Lorena Díaz González
Dr. Juan Manuel Rendón Mancha
Dr. Jorge Alberto Fuentes Pacheco
Dr. Audberto Reyes Rosas

Presidente:
Secretario:
Vocal:
Suplente:
Suplente:



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

JORGE HERMOSILLO VALADEZ | Fecha:2022-05-09 10:41:51 | Firmante

hn+nz2ew4bmUDsohkiBkOJMMRZNQDnTHUtsZLoA4HXYSb7FJU9T4zM+L2pq0MK/IDAtNWrxQRpV5Otlevt09YQF6alBotYWWWCaVf+WqBBGJrE8kbgagEzUmoXJqJoPipG75H VUy0t4dsrJue2aFA5e/mgE4ZO7LiAwwCHeFai2zxn4sS/ug2zcKKJ2O64DCm4hz38QgYqHej5qdwAqXc9y2LeV8KK43TrugQmxqq4EuLFSH171xOMKMTuyOt83k/K3Axeaz/BJY Awfawy1LQgUs8bazMVdkB/g9PbCq9RIR3cEEAGD+lb4jcX6SdfmpNYn7HQhLO1v7T6rAdFrhA==

JORGE ALBERTO FUENTES PACHECO | Fecha:2022-05-09 11:29:01 | Firmante

oTynrGDkLfwOMQd46fkdQ2xhy0tc4X7uPM/ka2ZEFIM7y/PmsKNSDeFQ3TvywPjYTD1oxFwS+2MmKvXph4cr1vqvOoNRKXS0H7LRR/U9ue8EGl5WzFGbE30gnkbhMpQclMF8 bav3Yl/KwLcMT74CPB4dvnLFS+6OUu23gMf1OZnj30QqOx1n0L23+nkZeKpKcF6wL+SuDB0T35c/NUtq2a1MZqPk8JOyPDHflkUbLvxqJDMITCf1m6mJoiKJW3fZbqClb9NiXw JwrzQaAg4dvkj7Zn6+0JtcS0JK9uAZTIsF8+OQwI4YZlzp2Blv9HjgTBwFZUnafNMntZGtjg==

JUAN MANUEL RENDON MANCHA | Fecha:2022-05-09 17:31:58 | Firmante

aC4xt5hUF7pKpz7u604wjFIRfiOXKdMQq3cl4LuxOUrNIR3QciUpknvaGkvieECv4QJrXE9j94ZGaqajNo9BgyiR7YD9u9s9X+E+hSiY7e2mVfcrwFhy70fle5PiJY0znaj8KS2+hKbrp xvKjrWwQtUzbaj+ULDxRAQWobulXdWkQB8ivrDKdD4cPCxKRMNW9e4j+OzzAJzCR3Pfl5tl6uSROsPYUkJnVpsvqx9kq/tk9zEUZJqt34xVx5RHZwYFjyZemUUfZXIcZnpi+IMah czsXg83V/kFuZNCuP8OSBNoly/sPc9nBKWECOP5f4FVDig0B5LY2250u7FaOSA==

LORENA DIAZ GONZALEZ | Fecha:2022-05-12 19:03:12 | Firmante

pba1qOnBh5QQKMOO4CB8t068c6aj9mJR/8g5FMzvEMe7QnskFhTGe8PtLXMFQZMJGKvgeFwEhgfgEkJtopobQIFFk9CatE+KAW6tV7cZLHISix3jnwTD+Qiz2t3hYmb4CUx BWdelibiNrLzNu8cl3/B+ZZroHR2RCtPOPvYgh2u5XFYcj/ua6LRk9B6CQVnrFWfanUwJa5wrlqij5/0k7ePFzLDIRaN5YGJacG4OhX3jsNTtekw9YkS1uqx3iQ8HC+sxui8Hg74vvy mvCI+Sj4ZYrzzYBqEaWlop3YJ7x3rXuFjWWHiS2uBlwg5ndZnCR3txl6HHc5ultzg==

AUDBERTO REYES ROSAS | Fecha:2022-05-17 22:32:06 | Firmante

pzHLZZYvXr58z+D0+N7L85U+X/7wgHyXXkTolPz3rc2kYVc1aFiXt0oOjlxSEoHfYnA8YtUjpWJt7SExJgX2sYYuOm8hmezk8UKvvp6ovOZxFaBeSjBlstqDnJbOKOZ/b+QtM4mN AvFZVwIk9e31GIQJNN9ILDNH3W0ocuANnoJ8BHxhGzZcyLrMjjFAWjwQWgdAEH7qnhRFGf6jHHWcGRWE4UF+uRsAnnZY7AFZqLtf3iUwhppWXQW53+gT5ZRTOWXxub phzzlbbKqKzGNY8q0BZOWArnoDPHZpUzqg1tACD0/k68h3x5W6LZzN3duEjfeive+y/rvjdf3Q==

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



2ga6wpYXN

<https://efirma.uaem.mx/noRepudio/VvFbzr9MOIJyG0RrhgfmVQq93MNdBjJt>

