

Proyecto Final

Image2 Circuit - Clasificador de elementos electrónicos usando una CNN.

Abstract / Resumen

En este proyecto buscaré investigar la fiabilidad y los resultados de usar una red neuronal profunda convolucional (CNN) para poder clasificar imágenes (tanto fotografías como imágenes esquemáticas creadas por computadora) de diversos elementos eléctricos (se piensa trabajar como mínimo con los 3 elementos básicos: resistores, capacitores e inductores) con un alto grado de confiabilidad (esperamos que dado un dataset balanceado de prueba obtener al menos un 85% de exactitud).

Para experimentar y llegar un modelo final en este trabajo se usará la librería Tensor Flow, y Google Colab (dadas las limitaciones de conexión y de los equipos de cómputo que tengo en el momento) para poder iterar, durante este y dado el largo proceso de entrenamiento que es común que dure de 20 minutos a una hora para 40 iteraciones sobre un dataset pequeño buscaré usar las capacidades de tensor flow para realizar esta tarea en paralelo así como medir diferencias en el tiempo de entrenamiento usando sólo CPUs, usando GPUs, y posiblemente TPUs.

Este trabajo servirá como pivote para poder empezar con el plan a más largo plazo de mi trabajo terminal, medio por el cual me titulare de la carrera de Ingeniería en Sistemas Computacionales de la ESCOM IPN, esto dado que estoy haciendo ambas carreras de manera simultánea (junto con Ciencias de la Computación en la Facultad de Ciencias), pero en la primera me encuentro ya en la recta final.

El objetivo final de dicho trabajo terminal será ocupar el modelo que estoy proponiendo y que apoyado con otros dos sistemas: uno que permita encontrar componentes de una imagen, basado en la arquitectura de YOLO y otro que permita encontrar como se relacionan (si existe conexión directa entre cualesquiera dos elementos), junto con el modelo que estoy creando que permita clasificar con fiabilidad a los componentes electrónicos para poder crear un grafo abstracto de conexión y usarlo para poder crear automáticamente esquemáticos a partir de la imagen de entrada o crear un archivo de simulación (por ejemplo para abrirlo en proteus) de manera automática.

Marco Teórico

Naturaleza y justificación

Cualquier disposición de las diversas fuentes de energía eléctrica junto con los diferentes elementos del circuito se denomina red eléctrica (circuito eléctrico / electrónico), los circuitos electrónicos son representados esquemáticamente con diferentes estándares tales como el ANSI o el IEEE.

Estos esquemas se ocupan en los diferentes simuladores que hay en el mercado los cuales nos permiten visualizar los resultados generados por dicha red.

Además, generalmente es necesario generar un reporte técnico y para realizarlo se utilizan desde los mismos simuladores hasta aplicaciones de dibujo para crear esquemas. Buscamos crear una herramienta de apoyo a la hora de digitalizar circuitos electrónicos minimizando el tiempo que el usuario dedica.

Justificación

Existen al menos 12 carreras de licenciatura en el Instituto Politécnico Nacional en las que tienen mínimo una unidad de aprendizaje relacionada directamente con circuitos electrónicos, al realizar diseños para dichas UA suele ser necesario simular los resultados o generar un reporte técnico con esquemas, este proceso donde se involucra la escritura a mano, el realizar la simulación y el reporte suele ser ocupar varias horas de trabajo, dependiendo de la experiencia con el simulador.

Generar un sistema para ayudar a la digitalización de dicho circuito nos minimizará tiempos de realización de prácticas y la posibilidad de realizar una simulación y obtener resultados en un menor tiempo.

Estado del Arte

A continuación, presentamos algunos de los trabajos de investigación y productos comerciales relacionados con la creación de una “netlist” y / o con la digitalización de circuitos electrónicos a partir de una imagen (generalmente una fotografía).

Software	Característica
CircuitSense Project	Es un proyecto (originalmente una aplicación solo para IOS) que convierte una fotografía de un circuito esquemático a una netlist. Requiere de un conjunto de datos de entrenamiento, los cuales no están incluidos y actualmente utiliza versiones de librería que ya no están disponibles por lo que resulta muy difícil usarlo actualmente.
NetlistViewer	Es una herramienta capaz de procesar una netlist de SPICE en texto plano y convertirla en un circuito gráfico.
Circuit recognition using netlist	Es un artículo que describe los pasos a seguir para poder transformar un circuito dibujado a mano a una netlist. Sin embargo, los autores no comparten la implementación.
Circuit Recognition with Deep Learning	Es una conferencia en donde se explica un método usando redes neuronales convolucionales para detectar sólo los elementos de circuitos, pero ya en un diagrama estandarizado.
Hand Drawn Optical Circuit Recognition	Es un artículo que explica el método de la red neuronal artificial para poder leer símbolos eléctricos de una foto de un circuito dibujado a mano. Tampoco comparten la implementación
Solución Propuesta	Será un sistema con un cliente que permita subir una imagen a un servidor donde se realizará todo el análisis de la imagen, se generará una predicción que se le regresará al usuario.

Redes Neuronales Convolucionales

Las redes neuronales son un subgrupo de la Inteligencia Artificial basado en un sistema de aprendizaje y procesamiento automático inspirado en el funcionamiento del sistema nervioso humano. Más concretamente, pertenecen al subgrupo del Machine Learning, una tecnología basada en crear sistemas que puedan aprender automáticamente, es decir, pueden descubrir patrones complejos enterrados en grandes conjuntos de datos sin la necesidad de interferencia humana.

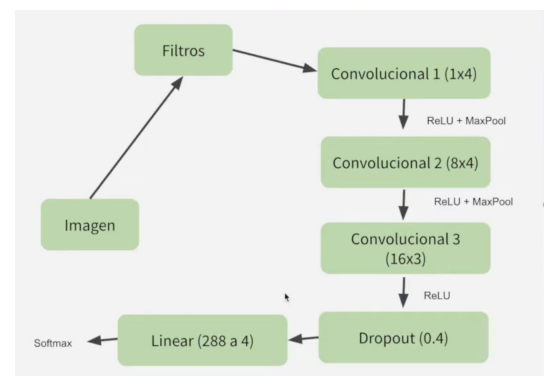
Las redes neuronales se diferencian por topología, dependiendo de las características con las que cuentan: perceptrón monocapa, la más sencilla; perceptrón multicapa, donde se desarrollan capas ocultas y es capaz de eliminar información irrelevante; Red Neuronal recurrente, donde cobra un papel importante la temporalidad y, con ella, se dota a la red neuronal de memoria y, por último, las Redes Neuronales Convolucionales; las más completas.

Cada parte de una red neuronal convolucional está entrenada para realizar una tarea, por lo que el entrenamiento de cada una de las partes se desarrolla de manera individual y se efectúa más rápido. Estas redes se utilizan en especial para el análisis de imágenes.

Desarrollo

Planeo usar como base el dataset que existe en Kaggle:

(<https://www.kaggle.com/xuowen/circuit-components/kernels>) eso si, hagase notar que no existe al parecer otro dataset disponible con las características necesarias para desarrollar el proyecto, por lo que se puede usar este como base apoyarnos con el mismo pero aumentarlo para tener mas datos (o mas clases incluso).



La idea para este proyecto será tomar las ideas que he estado realizando durante las últimas semanas en pytorch y pasarlos a TensorFlow con Keras, replicar la arquitectura que ya existe que es:

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=1, out_channels=8, kernel_size=4)
        self.conv2 = nn.Conv2d(in_channels=8, out_channels=16, kernel_size=4)
        self.conv3 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3)

        self.pool = nn.MaxPool2d(2)
        self.dropout = nn.Dropout2d(0.4)

        self.fc1 = nn.Linear(288, 4)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
```

```

x = F.relu(self.conv3(x))

x = self.dropout(x)
x = x.view(x.size(0), -1)

x = F.relu(self.fcl(x))

return x

```

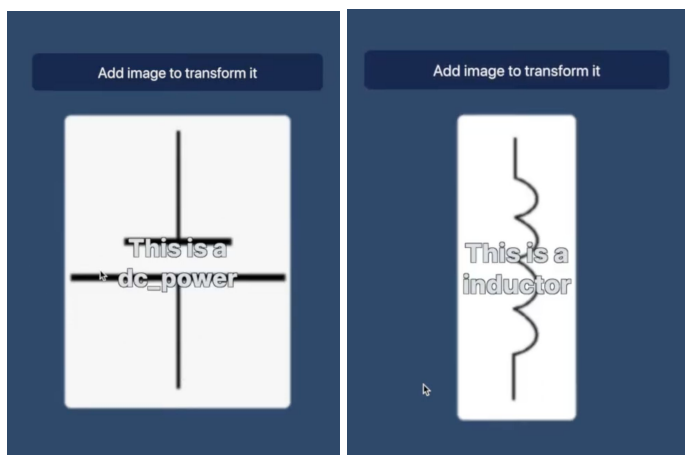
También, como mencionamos en la introducción y cito:

Para experimentar y llegar un modelo final en este trabajo se usará la librería Tensor Flow, y Google Colab (dadas las limitaciones de conexión y de los equipos de cómputo que tengo en el momento) para poder iterar, durante este y dado el largo proceso de entrenamiento que es común que dure de 20 minutos a una hora para 40 iteraciones sobre un dataset pequeño buscaré usar las capacidades de tensor flow para realizar esta tarea en paralelo así como medir diferencias en el tiempo de entrenamiento usando sólo CPUs, usando GPUs, y posiblemente TPUs.

Resultado

El resultado esperado es tener un modelo que pueda ser usado para predecir correctamente la clase a la que pertenece un componente electrónico con más de un 85% de exactitud (dado un dataset de pruebas balanceado y parecido al que usamos durante el entrenamiento pero si repetir ninguna imagen en ambos).

Esperamos si el tiempo lo permite también hacer deploy de este modelo usando un sistema cliente servidor muy básico o incluso usando tensorflow.js poder crear una pwa (progressive web app) que no requiera de un servidor para ejecutarse.



(Idea de como se vería al final)