

Projeto Integrado – DEL / POLI / UFRJ – 2022.1

Prof.: Casé e Juarez

Relatório do Projeto

PhotoSpice

Matheus Felinto

Lucas Tiné

Alan de Oliveira

29 de julho de 2022

Índice:

Introdução	1
Motivação	1
Metodologia	1
Limitações	1
Principais Ferramentas Utilizadas	2
Descrição do Funcionamento e Utilização	2
Envio e Análise da imagem	2
Processamento da imagem	3
Detecção de componentes e treinamento das redes	3
Geração da Netlist	4
Correção e Simulação dos Circuitos	5
Projetos Futuros	6

Introdução

Neste projeto tivemos como objetivo desenvolver uma aplicação que captura um circuito esquemático a partir de uma imagem, gera automaticamente uma netlist no padrão Spice e simula o circuito mostrando ao usuário as tensões nodais ou gráfico de tensão ou corrente dependendo do tipo de simulação e dos componentes presentes no circuito.

Motivação

Os simuladores SPICE atualmente são ótimas ferramentas ao se trabalhar com eletrônica, no entanto é preciso lidar com interfaces gráficas desses programas que podem não ser padronizadas e dependem do hardware e sistemas onde estão instaladas o que pode atrasar o processo de simulação como um todo. No projeto buscamos uma forma de simplificar o processo de simulação para fins de estudo para os alunos, usando um smartphone tirar uma foto do circuito que gostaria de simular e enviar para analisar. O projeto também teve inspiração em aplicações similares que fazem a solução de equações matemáticas a partir de fotos.

Metodologia

A partir de uma imagem do circuito capturada no celular e enviada pelo usuário para um servidor, o sistema irá processar a imagem modificando para que a mesma fique no padrão desejado para a geração da netlist. Depois que a imagem é processada usamos redes neurais para detectar os componentes e as conexões entre os mesmos. Para esta etapa treinamos três redes neurais sendo uma para a detecção geral de componentes e conexões, outra para a identificação dos números, unidades e múltiplos e outra para identificação das polaridades de determinados tipos de componentes. A seguir a netlist é gerada para a ferramenta usada para simular o circuito. Antes da simulação o usuário pode através da interface da aplicação fazer correções e escolher o tipo de simulação, a saber: ponto de operação e análise transiente, caso escolha pode escolher ainda de qual componente ou nó deseja ver o gráfico. Após as modificações e escolhas do tipo de simulação e componentes a serem analisados a simulação é realizada e os resultados exibidos pela aplicação. Todo o processo é intuitivo e rápido para exibir os resultados da simulação.

Limitações

Devido ao tempo disponível para o desenvolvimento do projeto, os componentes atualmente disponíveis para simulação são:

- Componentes passivos:
 - o Resistor
 - o Capacitor
 - o Indutor
- Diodos
- Fontes de tensão:
 - o Contínua
 - o Alternada

Futuramente desejamos incluir mais componentes, no entanto no momento o sistema já é funcional e pode simular uma ampla gama de circuitos como filtros passivos, retificadores de tensão, divisores de tensão, entre outros.

Principais Ferramentas Utilizadas

- Yolo: ferramenta de visão computacional para detecção e classificação de objetos em tempo real. No projeto foi utilizada para detectar os componentes, conexões e valores.
- PySpice: é uma interface open source Python com o simulador NGSpice
- Pytorch: open source framework de aprendizado de máquina
- Matplotlib: é uma biblioteca usada para visualizar os gráficos da simulação.
- Albumentations: usada para data-augmentation das imagens de circuitos e componentes usadas para treinar e testar as redes neurais.
- Vue.js e Bootstrap: front-end
- Flask: Backend

Entre outras como numpy, opencv e pycocotools.

Descrição do Funcionamento e Utilização

I. Envio e Análise da imagem

O usuário ao abrir aplicação web pode selecionar uma foto do esquemático e enviar para o servidor conforme pode ser visto na imagem do exemplo ao lado. Após a imagem ser enviada para o sistema deve-se clicar no botão analisar

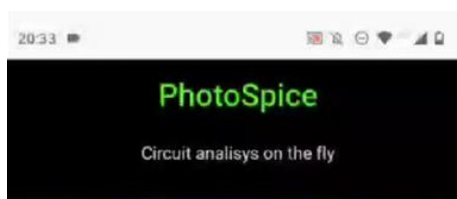


Fig. 1 - Interface da aplicação no celular



Fig. 2 – Interface da aplicação web computador

II. Processamento da imagem

Em seguida as imagens são binarizadas e o contorno do traçado é identificado para detectar as conexões entre os componentes.

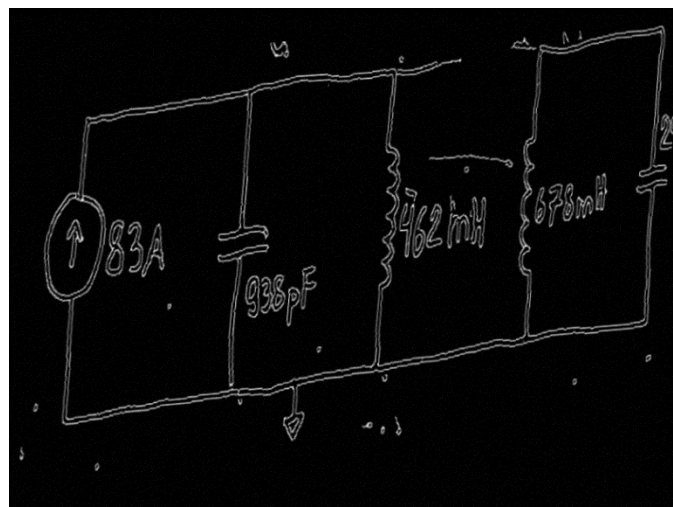
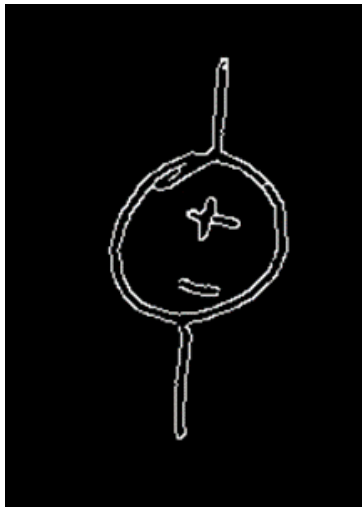


Fig. 3 – componentes binarizados

III. Detecção de componentes e treinamento das redes

- Data augmentation: para o treinamento e teste de todas as redes neurais usadas neste projeto, utilizamos a biblioteca Albumentations para criar uma variedade de imagens a partir de um pequeno conjunto de circuitos desenhado pelos membros do grupo ou retirados da internet.
- Detecção de componentes e conexões: foi utilizada a arquitetura de rede neural Yolo para a detecção dos componentes.

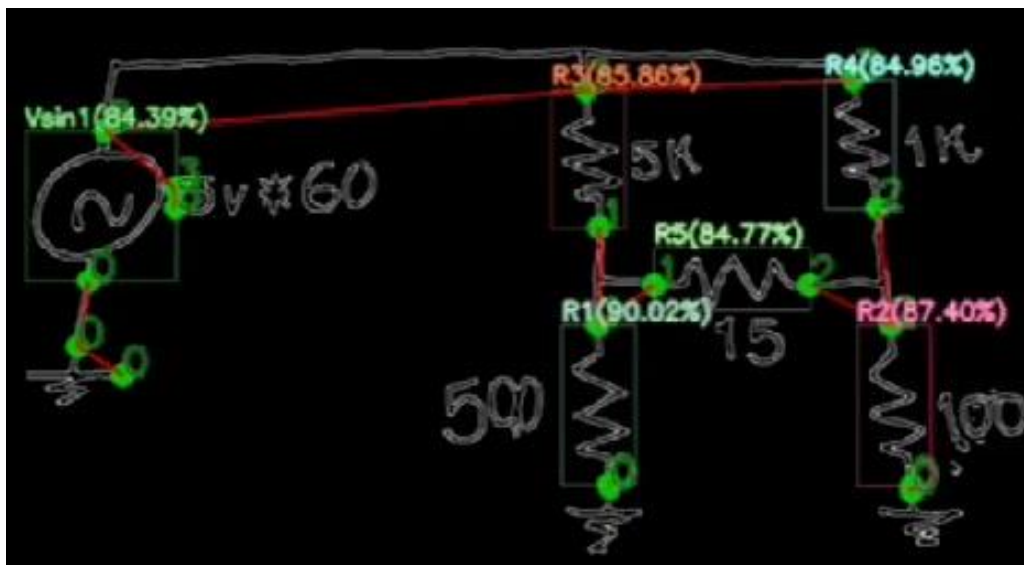


Fig. 4 - Detecção de componentes e conexões

- Detecção de números e unidades: também foi usada Yolo para o reconhecimento dos valores dos componentes as respectivas unidades múltiplas, como por exemplo: u para micro (10^{-6}).
- Polaridades: para identificar a polaridade dos terminais das fontes de tensão e dos diodos como precisávamos detectar os terminais positivo e negativo dos componentes, a abordagem usada para inferir as polaridades foi o uso de R-CNN keypoints models.

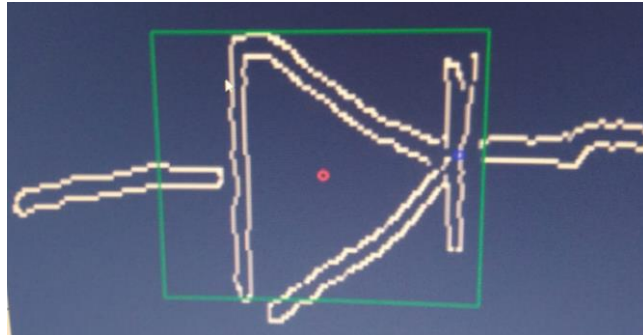


Fig. 5 - Detecção das polaridades do diodo

IV. Geração da Netlist

Após a identificação dos componentes, conexões, valores e polaridades dependendo dos componentes a aplicação gera automaticamente a netlist no padrão usado pelo PySpice para a seguir carregar a simulação de acordo com o tipo e parâmetros definidos pelo usuário na interface web.

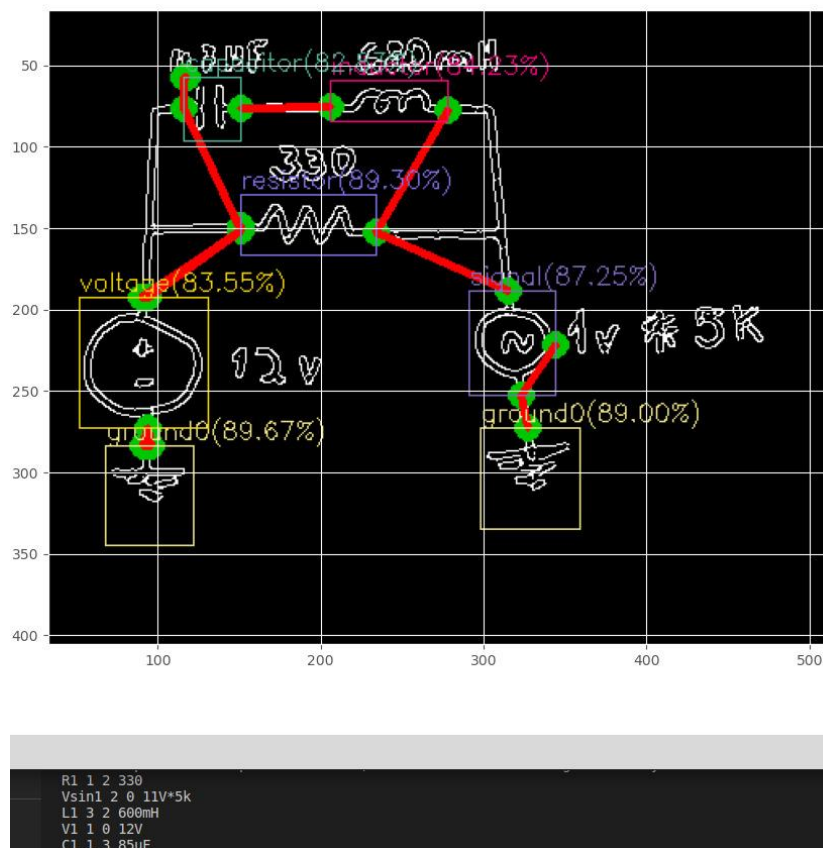
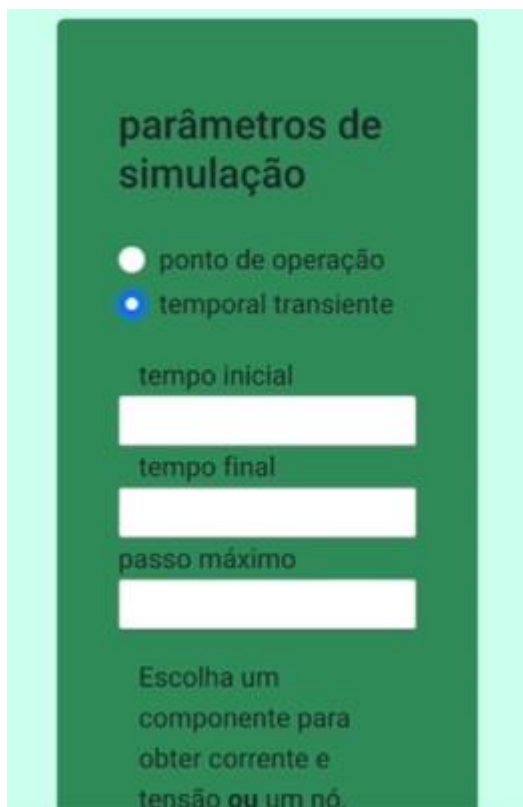


Fig. 6 – Netlist gerada para a simulação

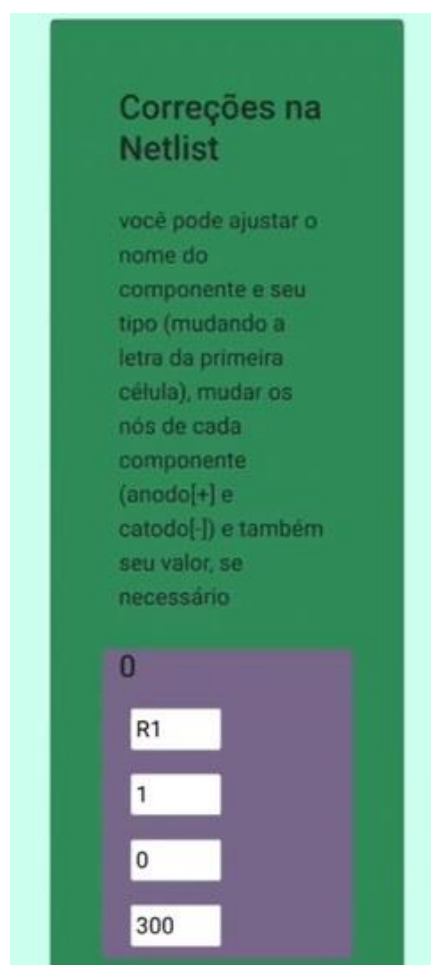
V. Correção e Simulação dos Circuitos

Depois que a netlist é gerada o usuário pode conferir se os valores reconhecidos pelas redes neurais estão corretos, caso seja necessário é possível modificar os parâmetros encontrados na netlist antes de simular o circuito. A aplicação trabalha com dois tipos de simulação, a saber: ponto de operação e análise transiente. O usuário seleciona o tipo de simulação desejado, caso escolha ponto de operação o programa exibe as tensões dos nós encontrados pelas redes neurais. Caso a análise transiente seja escolhida, também tem que selecionar se quer visualizar o gráfico de um determinado nó ou a corrente que circula por resistor, capacitor ou indutor. Também para a análise transiente é preciso informar os parâmetros: tempo inicial, tempo final e passo máximo.



The image shows a green rectangular form titled "parâmetros de simulação". It contains two radio buttons: "ponto de operação" (unselected) and "temporal transiente" (selected). Below these are three input fields labeled "tempo inicial", "tempo final", and "passo máximo". At the bottom, there is a text prompt: "Escolha um componente para obter corrente e tensão ou um nó,".

Fig. 7 - Parâmetros da análise transiente



The image shows a green rectangular form titled "Correções na Netlist". It contains a text prompt: "você pode ajustar o nome do componente e seu tipo (mudando a letra da primeira célula), mudar os nós de cada componente (anodo[+] e catodo[-]) e também seu valor, se necessário". Below this is a purple rectangular box containing four input fields with the values "R1", "1", "0", and "300".

Fig. 8 - Modificações na netlist

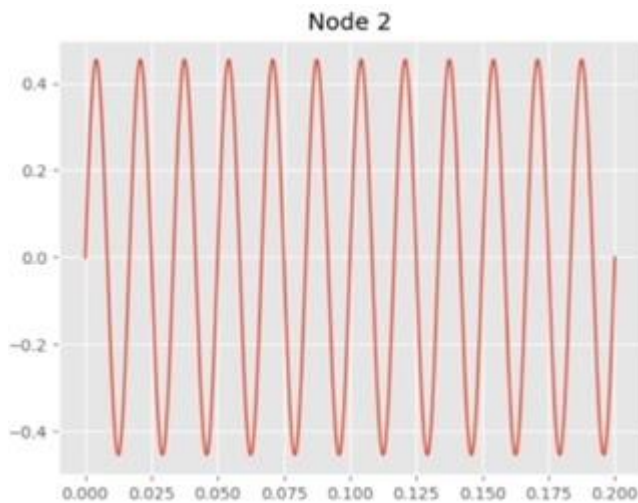


Fig. 9 - Gráfico da tensão nodal - Análise transiente

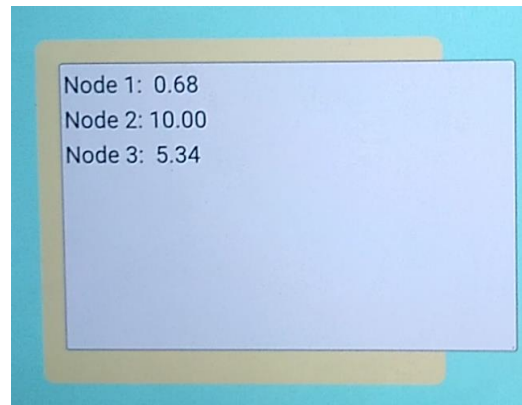


Fig. 10 - Tensões nodais – Ponto de Operação

Projetos Futuros

- I. Inclusão de mais componentes eletrônicos reconhecidos pelo sistema de detecção de componentes e simulação, como transistores bipolares, mosfets e amplificadores operacionais.
- II. A partir da netlist gerada para simular o circuito enviado gerar o diagrama esquemático para exibir acima dos nós ou componentes as correntes e tensões, conforme as imagens de exemplo abaixo:

