Issue 15, 2024

https://doi.org/10.23939/sisn2024.15.043

УДК 004.9

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПОБУДОВИ ВЕКТОРНИХ ДІАГРАМ ЕЛЕКТРИЧНИХ КІЛ

Євгеній Оніщенко[1], Дмитро Досин[2], Андрій Худий[3]

Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж, Львів, Україна
[1] E-mail: yevhenii.onishchenko.mnitm.2022@lpnu.ua, ORCID: 0009-0003-9460-4365
[2] E-mail: dmytro.v.dosyn@lpnu.ua, ORCID: 0000-0003-4040-4467
[3] E-mail: andrii.m.khudyi@lpnu.ua, ORCID: 0000-0003-2029-7270

Векторні діаграми – потужний інструмент для візуалізації та розуміння розподілу струму, напруги та потужності в електричних системах. Під час війни Росії проти України наша енергетична галузь стала дуже вразливою до атак ворога, а отже, потребує швидкого та ефективного відновлення. Фахівцям-енергетикам бракує програмних засобів для роботи з енергосистемою, а в період розвитку штучного інтелекту створити такі засоби не так важко.

Наприклад, знайомим спеціалістам часто доводиться будувати векторні діаграми – засоби для візуалізації та розуміння розподілу струму, напруги та потужності в електричних системах. Використовуючи поєднання фреймворків для роботи зі штучним інтелектом та створення графічної оболонки, можна досягти бажаного результату за кілька місяців, зробивши водночас корисну справу для нашої перемоги.

Серед запропонованих засобів для створення інтелектуальної системи побудови векторних діаграм розглянуто модель YOLO (на основі фреймворку Pytorch) та фреймворк QT. Роль штучного інтелекту полягає у розпізнаванні електричних елементів у колах та їхніх з'єднань один з одним. Створення інтерфейсу користувача не менш важливе, його вже у багатьох місцях впроваджено за допомогою QT.

Поки що не існує жодного спеціалізованого програмного засобу для вирішення проблем ручної побудови векторних діаграм, але запропоновані підходи вже використовують для його створення.

У системі застосовано модель Yolov5 для розпізнавання електричних елементів на схемі. Модель натренована на більш ніж 150 зображеннях та здатна розпізнавати нарисовані власноруч схеми. Розпізнавання запускається як окремий процес з основної програми, написаної мовою C++. Ця частина системи обробляє вхідні дані від Yolo, зберігає в зручному форматі, створює інтерфейс користувача та зображає результат у вигляді векторної діаграми.

Ключові слова: векторні діаграми; фреймворк; енергосистема; штучний інтелект; розпізнавання зображень; Yolo.

Вступ

У сучасному світі, де технології стрімко розвиваються, велику увагу приділяють розробленню інтелектуальних систем, які можуть оптимізувати та полегшити вирішення складних завдань у різних галузях науки та техніки. Одна з ключових галузей, де впровадження інтелектуальних технологій може забезпечити істотне покращення, – енергетика.

Центральним аспектом аналізу та оптимізації електричних систем є вміння ефективно розуміти та візуалізувати їхню структуру. Векторні діаграми електричних колів виявляються надзви-

чайно корисним інструментом для цього. Однак будувати їх і аналізувати іноді затратно та складно, особливо під час роботи із великим обсягом даних.

У статті розглянуто перспективи використання інтелектуальних систем для побудови векторних діаграм електричних кіл. Застосування штучного інтелекту дає змогу автоматизувати процес побудови, аналізу та оптимізації векторних діаграм, що сприяє підвищенню ефективності та точності роботи.

Постановка проблеми

Фахівці енергетичної сфери щодня ризикують життям, працюючи на підстанціях з високою напругою та забезпечуючи надійний трансфер електроенергії як в Україні, так і за її межами. Поспілкувавшись із фахівцями Львівської області, вдалося визначити програмні засоби, які вони використовують у своїй повсякденній діяльності, а також труднощі, з якими вони стикаються. Серед завдань, які досі виконують вручну, – побудова векторних діаграм. Незважаючи на важливість цих діаграм, ручне обчислення їх параметрів може бути дуже складним та часомістким завданням і, на жаль, відсутні програмні засоби для його автоматизації.

Аналіз останніх досліджень та публікацій

На жаль, обмежена кількість програмних засобів спеціалізується на створенні векторних діаграм. Звичайні редактори або програми для роботи з графікою, такі як Microsoft Visio, призначені для створення різноманітних типів діаграм, але не надають можливості побудови векторних діаграм за вказаними параметрами. Інші програми, такі як MATLAB та Scilab, можна використовувати для відображення, але не для створення векторних діаграм, і вони потребують створення проєкту в їхньому середовищі, що доволі складно для звичайного фахівця-енергетика.

Інтернет має лише один ресурс, призначений саме для побудови векторних діаграм за заданими значеннями, але його створили розробники з країни-агресора [1]. Крім того, цей інструмент обмежений у функціональності та потребує постійного підключення до мережі. Енергетики розповідають, що на підстанціях рідко є доступ до інтернету або сигнал дуже слабкий. Отже, можна стверджувати, що сьогодні немаєує програмних засобів для побудови векторних діаграм.

Створення інтелектуальної системи побудови векторних діаграм передбачає розроблення моделі штучного інтелекту, її тренування та тестування. Розвиток штучного інтелекту в енергетичній галузі України не надто стрімкий. Цю проблему та підходи до її вирішення висвітлено в аналітичній доповіді на тему “Штучний інтелект в енергетиці” [2].

Це літературне джерело містить огляд можливостей використання систем штучного інтелекту (ШІ) в енергетиці, зосереджуючись на таких ключових аспектах, як взаємодія з клієнтами, управління мережами, управління мікромережами, віртуальними електростанціями та новими бізнес-моделями на ринку електроенергії.

За результатами опитування, яке провела компанія Siemens у 2019 р. [3] серед понад 500 керівників енергетичних компаній, виявлено обмежений обсяг використання ШІ у різних напрямках. Найпоширеніші напрями застосування штучного інтелекту: прогнозування обслуговування активів, покращення автоматизації машин та обладнання, оптимізація процесів, програмного забезпечення чи інструментів.

У цьому проєкті використання штучного інтелекту полягає в розпізнаванні об’єктів на зображенні, тому доцільно порівняти різні рішення в цьому напрямі. Зокрема порівняння алгоритмів розпізнавання об’єктів здійснено у статті “Overview of two-stage object detection algorithms” [4]. Автори зазначають, що методи виявлення об’єктів переважно використовують одноетапні та двоетапні алгоритми виявлення об’єктів. Поширені одноетапні алгоритми – YOLO, SSD, CornerNet тощо. Дослідники порівняли також основні особливості цих алгоритмів:

Алгоритм YOLO підвищує точність, швидкість та знижує кількість промахів і помилкових виявлень. У SSD вища продуктивність виявлення, що забезпечує дві переваги – у режимі реального часу та високої точності. Однак SSD ефективніше виявляє великі об’єкти, але менш ефективний для

малих об'єктів. Алгоритм виявлення CornerNet використовує ключові точки для перетворення кадру виявлення, що дає змогу передбачити ключові моменти об'єктів безпосередньо.

Двоетапний алгоритм хоча й точніший, але складніший. Він складається з двох етапів: спочатку виконують попередні тести, і лише після цього генерують регіони інтересу (RoI); на другому етапі виконується алгоритм регіональної класифікації та уточнення розташування на RoI, створених на попередньому етапі. Популярні двоетапні алгоритми – Faster R-CNN, R-FCN, FPN тощо.

Швидший алгоритм R-CNN використовує RPN на основі Fast R-CNN, інтегруючи їх у повну мережу, що уможливлює наскрізне навчання та підвищує швидкість. Однак обчислення кожного RoI всередині кожного шару згортки може бути неефективним, тому швидкість виявлення Faster R-CNN все ще недостатньо висока для реального часу. Алгоритм R-FCN використовує повну структуру згорткової мережі, що уможливлює спільне використання параметрів та забезпечує високу точність виявлення і швидкість. Алгоритм FPN використовує звичайні моделі CNN для ефективного вилучення характеристик різних розмірів у зображеннях і покращення передавання та виведення інформації про зображення.

Розглядаючи конкретніші приклади, можна виокремити роботу "Visual Object Detection and Tracking using YOLO and SORT" [5], де використано YOLO модель. Дослідники створили спеціальний набір даних, який містить кількасот зображень із шести класів (людина, автомобіль, вантажівка, автобус, велосипед і мотоцикл) для навчання YOLOv3. До цього YOLOv3 було попередньо навчено на наборі даних MS COCO з 80 класами. Модель навчали протягом 320 епох. Всі 800 зображень були вручну анотовані за допомогою інструменту LabelImg у форматі YOLO, з них 200 зображень використали для перевірки.

Після завершення анотування зображень та позначення міток всі дані розміщують у каталозі. Цю інформацію передають як параметри або код у головний файл і за допомогою бібліотеки PyTorch [7] модель YOLOv3 навчається. Файл з вагами – це остаточний результат навчання, який буде використовуватися для виявлення об'єктів у моделі.

Вхідне відео проходить через систему; на кожному кадрі обчислюються обмежувальні рамки та ідентифікатори класів за допомогою детектора об'єктів YOLO. Потім виявлене передають до трекера SORT. Кожній обмежувальній рамці присвоюють унікальний ідентифікатор, а у разі втрати об'єкта трекер призначає новий ідентифікатор та починає відстежувати новий об'єкт.

Такий підхід до використання штучного інтелекту доцільний для розроблення системи побудови векторних діаграм.

Формулювання цілі статті

Мета статті – аналіз автоматичних засобів для роботи в енергетичній галузі України. Важливо розуміти, що під час війни швидке та якісне вирішення проблем на енергетичних підстанціях, ТЕЦ, АЕС, ВЕС, ГЕС є критичним фактором для відновлення справедливого миру. Для повної заміни трансформатора необхідно 3–12 місяців, ураховуючи стадії проєктування, налагодження, тестування та введення в експлуатацію. Автоматизація процесу побудови векторних діаграм може забезпечити економію часу, але для фахівців-енергетиків все ще залишається багато роботи. Зважаючи на прогрес технологій машинного навчання, багато функцій може взяти на себе штучний інтелект.

Виклад основного матеріалу

Побудова векторних діаграм – важливий етап аналізу та проєктування систем змінного струму та електричних кіл. Векторні діаграми допомагають інженерам та дизайнерам візуалізувати фазові взаємовідносини, амплітуди, імпеданси та інші параметри, що характеризують змінний струм та напругу [6]. Детальніше застосування векторних діаграм наведено нижче:

- Аналіз та розрахунок фазових параметрів. Векторні діаграми використовують для аналізу та визначення фазових параметрів у електричних системах. Це передбачає визначення фазових кутів, фазових напруг та струмів.

- Розрахунок активної та реактивної потужності. Векторні діаграми візуалізують фазові зміщення та взаємодії між напругою і струмом. Активну та реактивну потужність визначають на основі геометричних відносин між цими векторами.
- Управління фактором потужності. Векторні діаграми візуалізують фазові відносини та кути між векторами напруги та струму. PF можна розглядати як проєкцію струмового вектора на вектор напруги. Векторні діаграми використовують, щоб визначити оптимальні розмір та тип компенсаційних пристроїв для досягнення бажаного фактора потужності. Використання автоматизованих систем, які виявляють зміни у факторі потужності, дає змогу ефективно управляти великими електричними мережами та підтримувати стійкий фактор потужності.
- Проєктування та аналіз електричних мереж. Векторні діаграми використовують для проєктування та аналізу електричних мереж, ураховуючи трифазні системи, що типово для багатьох енергетичних систем. Використовуючи діаграми, можна оптимізувати розташування компонентів у мережі для досягнення максимальної ефективності та стійкості, а також для визначення необхідності й ефективності резервних ліній та елементів.
- Визначення комплексного імпедансу. Векторні діаграми допомагають визначити комплексний імпеданс у системах змінного струму, що важливо для оцінки взаємодії між напругою та струмом.
- Аналіз та візуалізація загального стану системи. Векторні діаграми надають інженерам можливість візуалізувати та аналізувати стан електричних систем у комплексі, що корисно для виявлення проблем та розроблення стратегій управління. Векторні діаграми можна використовувати для візуалізації та аналізу тривалості збоїв у електричних системах.

Розвиток програмних засобів для автоматизації цього процесу стає надзвичайно важливим для полегшення і прискорення роботи інженерів та дизайнерів. Приклад топографічних діаграм напруг і струмів наведено на рис. 1.

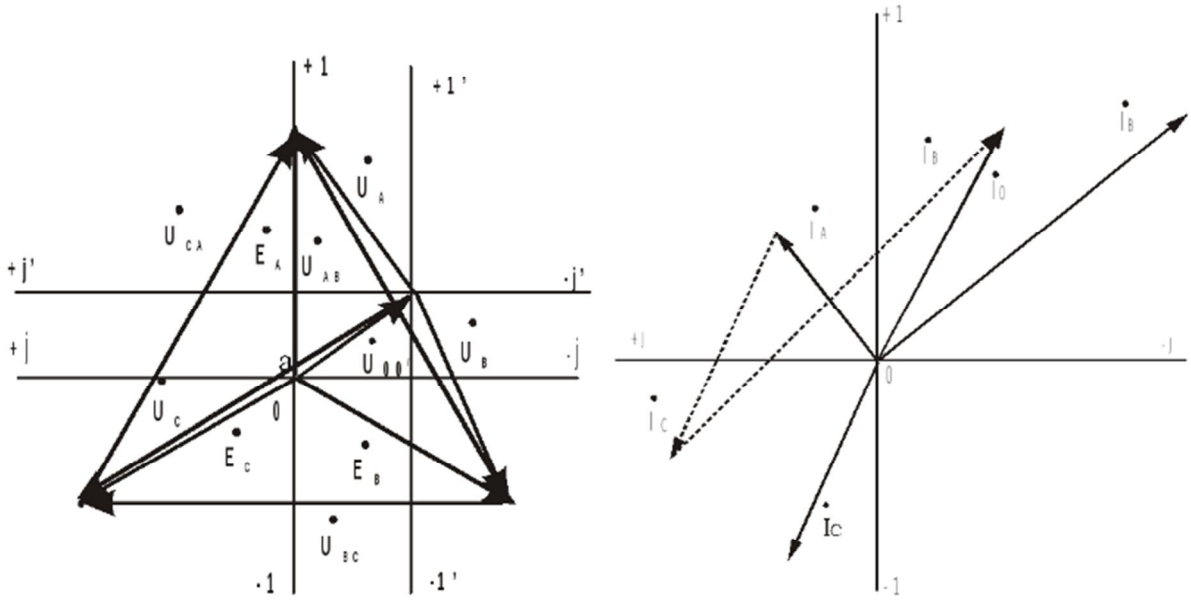

Рис. 1. Приклад побудови векторних діаграм

Програмний застосунок, створений на основі ШІ, здатен значно полегшити та пришвидшити роботу з векторними діаграмами електричних кіл, зменшити ймовірність помилок та зробити аналіз і проєктування доступнішими та ефективнішими завдяки використанню інтелектуальних методів.

Важливою вимогою для побудови векторних діаграм є не просто графічне зображення векторів струмів та напруг за заданими координатами, а й обчислення цих величин на основі схеми

електричного кола. Отже, ШІ може бути використаний для розпізнавання зображення схеми і автоматичного заповнення вхідних параметрів.

Концептуальна модель системи складається із таких елементів:

- Модуль розпізнавання розпізнає об'єкти електричного кола на зображенні та зберігає результат у вигляді іншого зображення.
- Користувацький інтерфейс містить елементи управління та візуалізації; користувач може вводити дані, вибирати параметри та спостерігати за векторними діаграмами.
- Модуль опрацювання даних відповідає за приймання та опрацювання даних, які ввів користувач. Цей модуль містить алгоритми аналізу та обчислення параметрів для побудови векторних діаграм.
- Модель зберігає інформацію про параметри фазових векторів та попередні результати розрахунків.
- Графік відповідає за відображення векторних діаграм на підставі опрацьованих даних.
- Модуль взаємодії із користувачем забезпечує обмін даними між інтерфейсом та модулем опрацювання даних. Реагує на команди та запити, які вводить користувач.
- Модуль валідації перевіряє правильність та адекватність введених користувачем даних та видає повідомлення про помилки.

У розробленому застосунку використано модель YOLOv5 для розпізнавання елементів електричних схем. YOLO (You Only Look Once) – це популярний алгоритм для об'єктного визначення в зображеннях та відео. Основна ідея YOLO полягає в тому, що він використовує один класифікатор для всіх об'єктів, що належать певній області зображення. Такий підхід робить YOLO дуже швидким і придатним для використання у реальному часі.

Зупинившись на версії YOLOv5, отримуємо всі переваги – достатньо високу швидкодію, простішу і зрозумілішу архітектуру самої моделі YOLO. Найбільшою перевагою YOLOv5 порівняно із попередниками є відкритий код. Це дає змогу детальніше ознайомитися з її архітектурою, залучити більшу аудиторію, яка згодом може стати чудовим інструментом підтримки продукту.

За допомогою YOLO було натреновано модель машинного навчання на більше ніж 150 зразках електричних схем. Тренування зображено на рис. 2. Вручну було вказано назви всіх елементів на схемі. Елементів на рисунках всього п'ять – джерело напруги, котушка індуктивності, резистор, діод та конденсатор. Цих елементів достатньо для побудови векторних діаграм трансформаторів, генераторів та як варіант просто розрахунку значень загальної напруги, струму, опору в колі для розв'язання інших завдань.

Процес розпізнавання є частиною всієї системи і запускається окремо з головної програми, написаної мовою C++. Ця програма виконує функції графічного інтерфейсу, виклику API розпізнавання елементів, розрахунку значень електричних величин, побудови графіка векторної діаграми.

Нині функціональні можливості розробленого проєкту такі:

- Розпізнавання електричних елементів на схемі: резисторів, джерел напруги, індуктивних котушок, діодів, конденсаторів.
- Розпізнавання з'єднання елементів.
- Забезпечення зручного інтерфейсу користувача.
- Побудова векторної діаграми за значеннями, які задав користувач.
- Швидкий розрахунок електричних величин (струму та напруги) в електричному колі з одним джерелом напруги та послідовним з'єднанням елементів.
- Побудова векторної діаграми для вищевказаних величин.
- Підтримка використання трьох фаз під час побудови діаграми.

Інформаційна система створена так, щоб вона була відкритою до збільшення цих функціональних можливостей. Архітектура ґрунтується на декількох шаблонах проєктування, зокрема підходу MVC.

З огляду на важливість створення інтерфейсу користувача, програма основана на QT фреймворку [8]. QT дає розробникам змогу створювати ефективний та кросплатформний користувацький інтерфейс. Він також має велику кількість готових компонентів, що істотно полегшує розроблення. Фреймворк має вбудований MVC (Model-View-Controller) підхід. Цей підхід є шаблоном проєктування, який дає змогу розділити компоненти програмного засобу на три окремі частини: модель (Model), представлення (View) та контролер (Controller). Кожна з цих частин виконує свою функцію та взаємодіє з іншими частинами програмного засобу через строго визначений інтерфейс. Також QT ґрунтується на принципі signal-slot, що дозволяє створювати власні засоби опрацювання сигналів (наприклад, можна прописати дії, які будуть виконуватися після натискання кнопки).

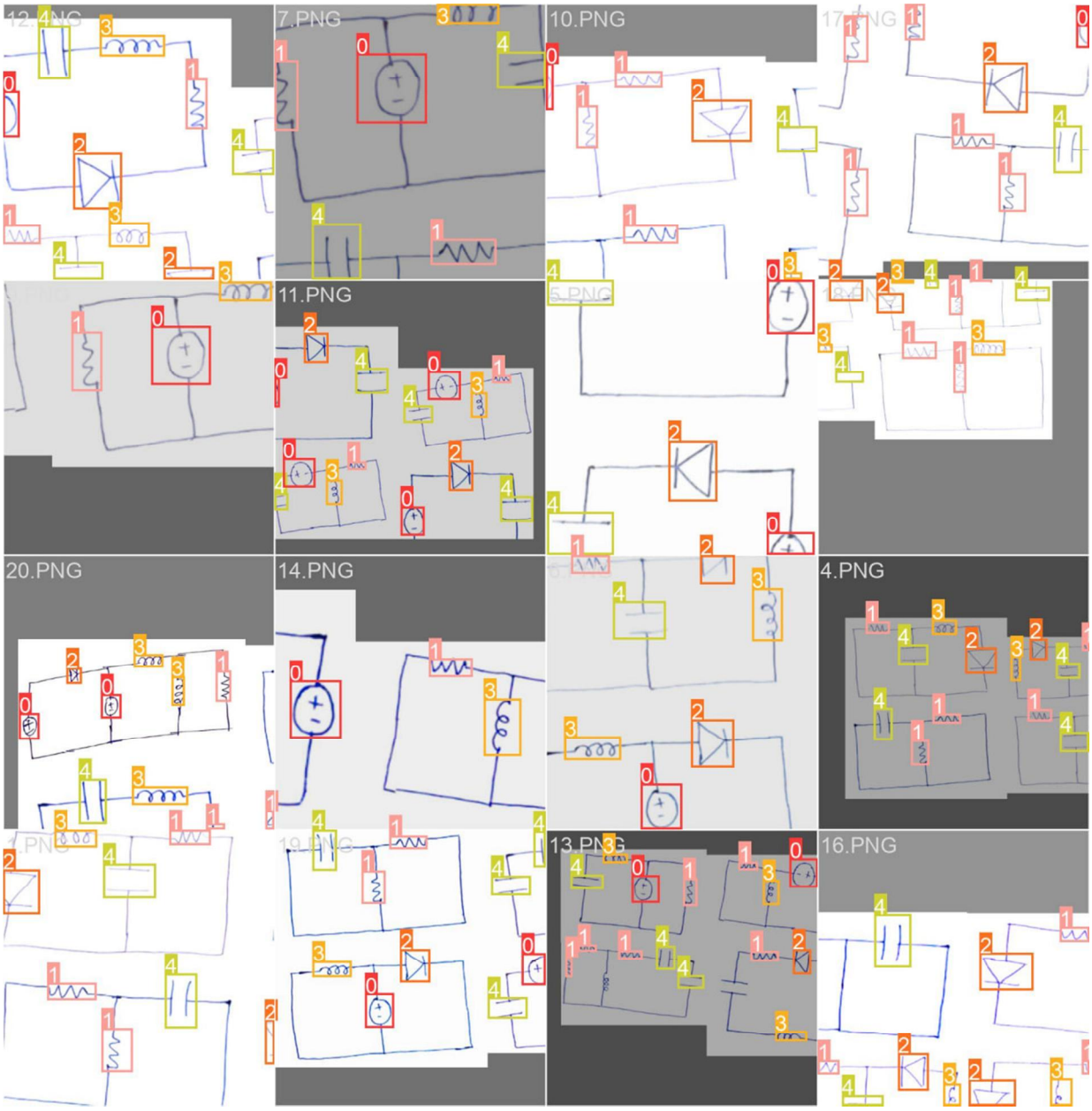

Рис. 2. Приклад тренування моделі YOLO

Qt фреймворк використовують у багатьох великих проєктах та корпоративних програмах, охоплюючи різноманітні галузі від програмного забезпечення для вбудованих систем до настільних

застосунків та мобільних додатків, наприклад, у VirtualBox, MATLAB (за допомогою MATLAB Engine API для C++), VLC Media Player, Google Earth, Telegram Desktop.

Структуру C++ частини системи зображено на діаграмі класів (рис. 3, 4).

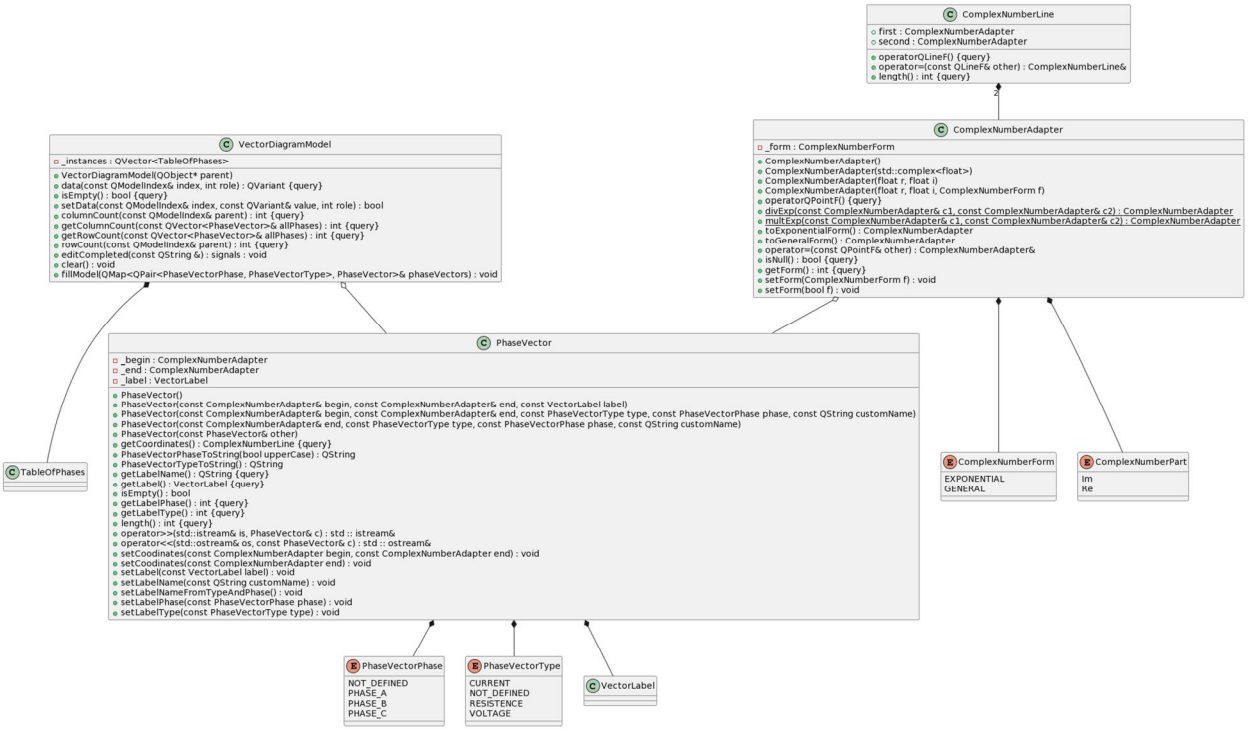

Рис. 3. Діаграма класів частини типів даних та моделі

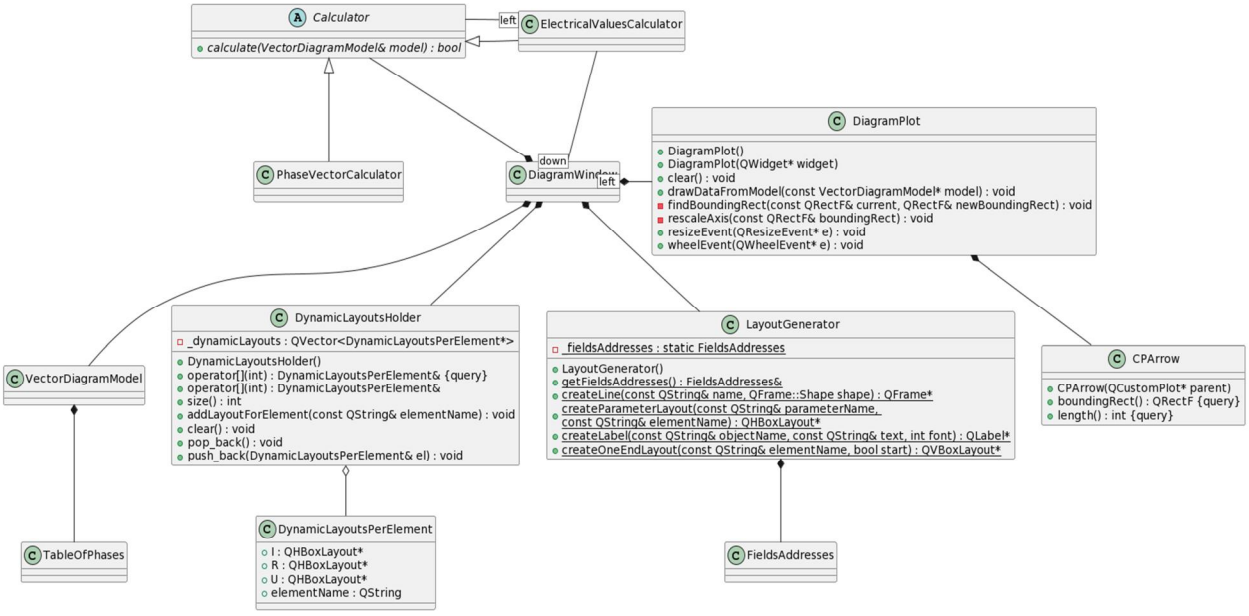

Рис. 4. Діаграма класів частини розрахунків та відображення (спрощена)

Тут використано власну структуру даних PhaseVector для зберігання значень одного вектора, який має координати початку, кінця, які є комплексними числами, а також фазу, тип, назву. Усі вектори зберігаються в VectorDiagramModel – таблиці векторів, створеній за допомогою фреймворка QT.

Побудовою графіка займається клас DiagramPlot, успадкований від QCustomPlot [9]. QCustomPlot дає змогу легко створювати різні типи графіків, такі як лінійні графіки, стовпчасті діаграми, точкові графіки, криві Безьє, спектрограми та багато інших. Користувач може взаємодіяти з графіками, наведенням на точки, масштабуванням та переміщенням. За допомогою QCustomPlot можна налаштовувати колір, тип лінії, розміри точок і багато іншого для кожного набору даних.

Дані з моделі видобувають та зображають на графіку, який, по суті, є тільки елементом подання діаграми, за допомогою класу DiagramView. Зв'язок між DiagramView та VectorDiagramModel реалізовано через клас DiagramWindow.

Також DiagramWindow відповідальний за створення графічного інтерфейсу (рис. 5).

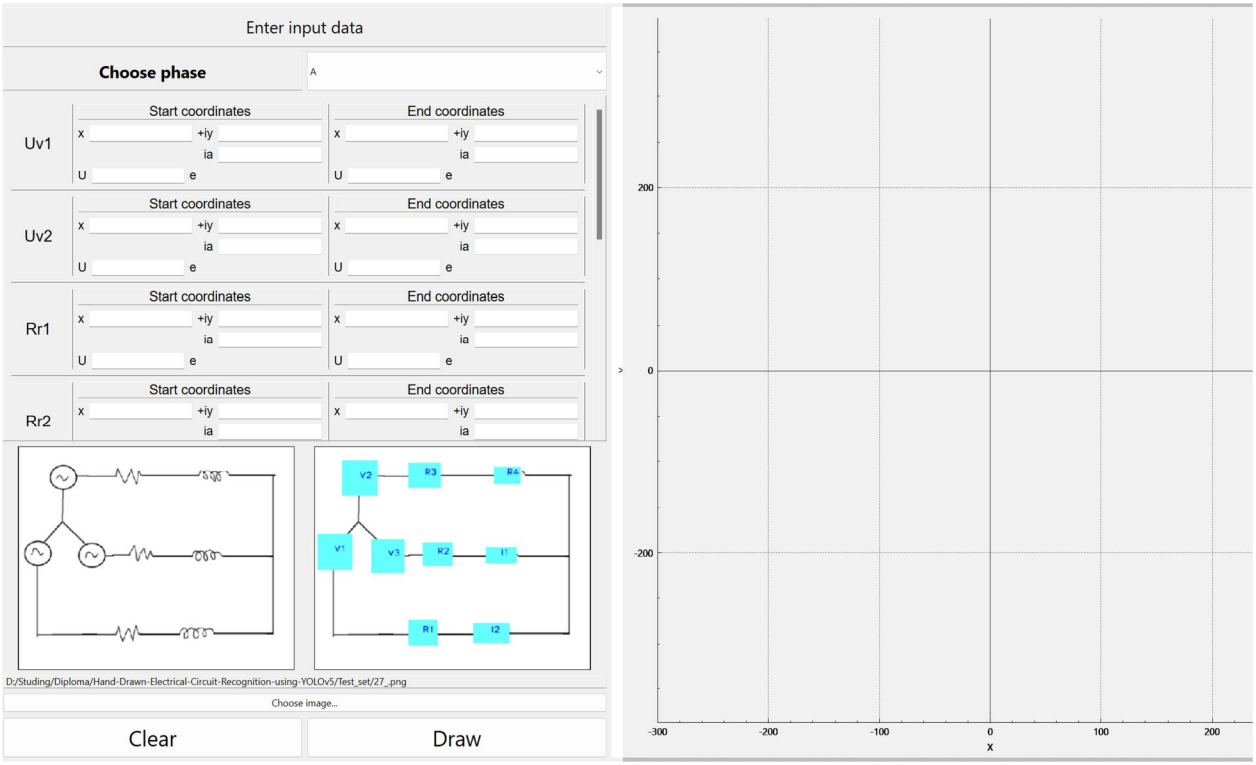

Рис. 5. Графічний інтерфейс системи

Є можливість завантажити власне зображення, яке опрацює описаний вище окремий процес розпізнавання елементів кола. Результат виводиться одразу ж поруч із оригінальним зображенням.

Усі необхідні поля генеруються автоматично для кожного елемента схеми, щоб користувач міг вказати необхідні параметри. Забезпечено прокручування полів параметрів, адже їх може бути доволі багато.

Взаємодію полів, кнопок та діаграми реалізовано за допомогою сигнал-слот підходу, що є своєрідним шаблоном Спостерігач в QT фреймворку. Він використовується для опрацювання подій та комунікації між об'єктами в програмі. Цей підхід дає змогу зв'язувати сигнали, згенеровані в одному об'єкті (наприклад, кнопка натиснута), зі слотами (спеціальними функціями), які виконуються в іншому об'єкті або контексті. Сигнали – це спеціальні методи-члени класу, які викликають у разі певних подій. Наприклад, якщо натиснути на кнопку, вона може відправляти сигнал clicked(). Слоти – це методи, виконувані відповідно до сигналів, які вони приймають. Наприклад, якщо є слот onButtonClicked(), він може бути пов'язаний із сигналом clicked() від кнопки.

У системі побудови векторних діаграм такий механізм використано зокрема у методах onChooseImageButtonClicked(), onDrawBtnClicked(), onStartGenXEditTextEdited(), onStartGenYEditTextEdited() і подібних.

Особливістю інформаційної системи є автоматичне перетворення комплексного числа з алгебраїчної форми у степеневу та навпаки відразу ж після введення. Передбачено підтримку простого зображення векторних діаграм на основі введених даних без завантаження зображення схеми. Простий приклад подано на рис. 6.

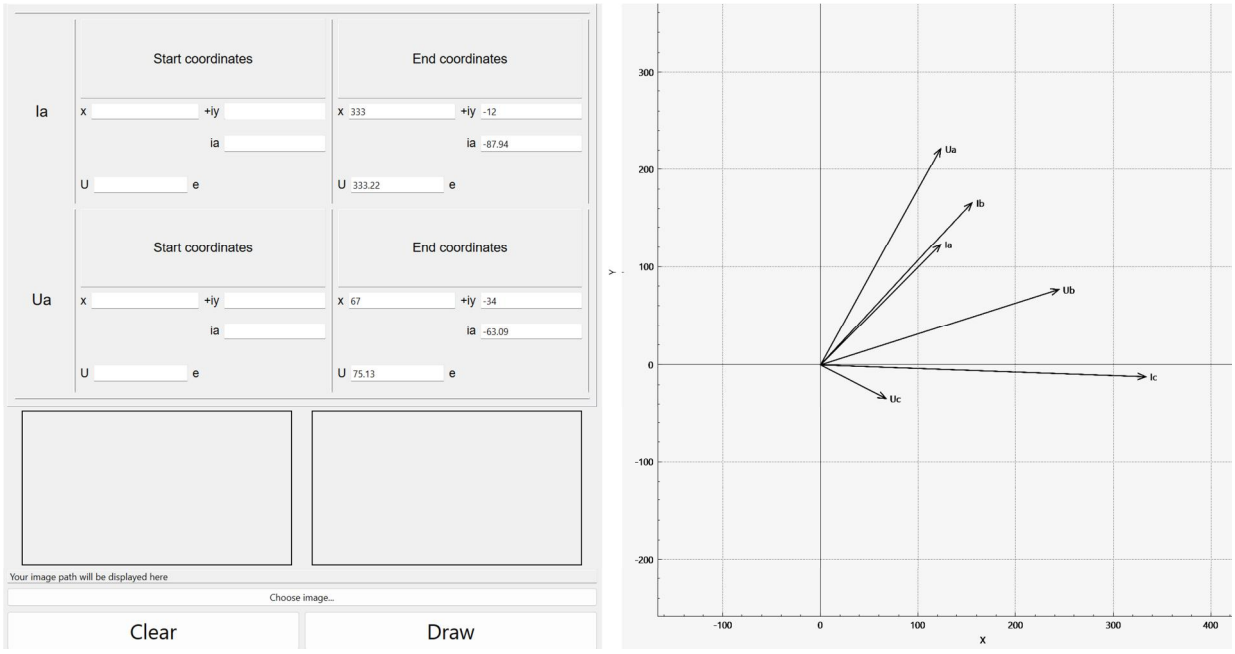

Рис. 6. Створення векторної діаграми за параметрами, які задав користувач

Створення застосунку можна простежити на ресурсі GitHub [10].

Точність розпізнавання елементів електричної схеми вказано в таблиці [11].

Точність класифікації алгоритму YOLOv5 [11]

Клас	Точність	Повнота	F1-міра
Джерело напруги	100	100	100
Конденсатор	96,30	96,30	96,30
Котушка індуктивності	97,44	97,44	97,44
Резистор	96,55	100	98,24
Діод	100	100	100
Середнє	98,06	98,75	98,40

F-міра є однією із метрик точності тестування. Її обчислюють на основі точності та повноти тесту. Точність визначають як відношення правильно встановлених позитивних результатів до всіх визначених позитивних результатів, разом із визначеними неправильно, тоді як повнота – як відношення правильно визначених позитивних результатів до всіх позитивних зразків, які мали б бути визначені як позитивні. Формулу для обчислення F1-міри наведено нижче:

Висновки

Використання штучного інтелекту в енергетичній сфері для створення векторних діаграм, безперечно, є важливим та перспективним напрямом розвитку. Протягом останніх десятиліть в енергетичній галузі відбулись істотні зміни, такі як збільшення складності систем, зміни джерел

енергії та зростання відповідальності за екологічну сталість. У цьому контексті штучний інтелект стає важливим інструментом для вирішення багатьох завдань, пов'язаних зі створенням стійких, ефективних та екологічно чистих енергетичних систем.

Сучасні системи штучного інтелекту, зокрема машинне навчання та нейронні мережі, можуть аналізувати та прогнозувати фазові взаємодії, імпеданси та інші параметри на основі великих обсягів даних. Це дає змогу інженерам і дослідникам ефективно використовувати векторні діаграми для оптимізації систем енергетики, зниження викидів та розроблення стійких та інтелектуальних мереж.

Важливо підтримувати баланс між автоматизацією та контролем, особливо в галузі, де надійність та безпека мають критичне значення. Штучний інтелект повинен використовуватися як інструмент для полегшення та підтримки прийняття рішень, а не для заміщення людського досвіду та експертності.

Список літератури

1. Voronov, P. (2019). Building of vector diagrams. https://faultan.ru/vector_diagram_faq/
2. Суходоля, О. М. (2022). *Штучний інтелект в енергетиці* [Аналітична доповідь]. Національний інститут стратегічних досліджень. https://doi.org/10.53679/NISS-analytrep.2022.09
3. Siemens company (2020). *Next-Gen AI in Energy: A Tool for Transition* [Опитування компанії Siemens щодо використання ШІ в різних напрямах]. Siemens AG in Cooperation With Longitude Research Ltd. https://assets.new.siemens.com/siemens/assets/api/uuid:fef90d09-6876-4510-b29b-bb6d60374793/siemens-next-gen-industrial-ai-energy-sector.pdf
4. Lixuan D., Rongyu Z., Xiaotian W. (2020). *Overview of two-stage object detection algorithms*. Journal of Physics: Conference Series. https://doi.org/10.1088/1742-6596/1544/1/012033
5. Grishma S. (2019). Visual Object Detection and Tracking using YOLO and SORT. *International Journal of Engineering Research & Technology (IJERT).* ISSN: 2278-0181. https://www.ijert.org/research/visual-object-detection-and-tracking-using-yolo-and-sort-IJERTV8IS110343.pdf
6. William H. Hayt, John A. Buck. (2012) Engineering Electromagnetics, eight edition. С. 1-26.
7. PyTorch фреймворк (2017). https://pytorch.org/
8. QT фреймворк (1991). https://www.qt.io/
9. Документація для класу QCustomPlot (2021). https://www.qcustomplot.com/
10. Репозиторій GitHub з прототипом проєкту. https://github.com/Aratimaru/VectorDiagram/tree/master
11. Rohith, R., Mahesh, R. (2021). Hand-Drawn Electrical Circuit Recognition using Object Detection and Node Recognition. https://doi.org/10.48550/arXiv.2106.11559

References

1. Voronov, P. (2019). Building of vector diagrams. https://faultan.ru/vector_diagram_faq/
2. Sukhodolia, O. M. (2022). *Artificial intelligence in energy* [Analytical report]. National institute of strategic research. https://doi.org/10.53679/NISS-analytrep.2022.09
3. Siemens company (2020). *Next-Gen AI in Energy: A Tool for Transition* [Survey of the Siemens company regarding the use of AI in various areas]. Siemens AG in Cooperation With Longitude Research Ltd. https://assets.new.siemens.com/siemens/assets/api/uuid:fef90d09-6876-4510-b29b-bb6d60374793/siemens-next-gen-industrial-ai-energy-sector.pdf
4. Lixuan D., Rongyu Z., Xiaotian W. (2020). *Overview of two-stage object detection algorithms*. Journal of Physics: Conference Series. https://doi.org/10.1088/1742-6596/1544/1/012033
5. Grishma S. (2019). Visual Object Detection and Tracking using YOLO and SORT. *International Journal of Engineering Research & Technology (IJERT).* ISSN: 2278-0181. https://www.ijert.org/research/visual-object-detection-and-tracking-using-yolo-and-sort-IJERTV8IS110343.pdf
6. William H. Hayt, John A. Buck. (2012) Engineering Electromagnetics, eight edition. No. 1-26.
7. PyTorch framework (2017). https://pytorch.org/
8. QT framework (1991). https://www.qt.io/
9. QCustomPlot class documentation (2021). https://www.qcustomplot.com/
10. GitHub repository with the project prototype. https://github.com/Aratimaru/VectorDiagram/tree/master

11. Rohith R., Mahesh R. (2021). Hand-Drawn Electrical Circuit Recognition using Object Detection and Node Recognition. https://doi.org/10.48550/arXiv.2106.11559

INTELLIGENT SYSTEM OF CONSTRUCTING VECTOR DIAGRAMS OF ELECTRICAL CIRCUITS

Yevhenii Onishchenko[1], Dmytro Dosyn[2], Andrii Khudyi[3]

Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine,
[1] E-mail: yevhenii.onishchenko.mnitm.2022@lpnu.ua, ORCID: 0009-0003-9460-4365
[2] E-mail: dmytro.v.dosyn@lpnu.ua, ORCID: 0000-0003-4040-4467
[3] E-mail: andrii.m.khudyi@lpnu.ua, ORCID: 0000-0003-2029-7270

Phasor diagrams are a powerful tool for visualizing and understanding the distribution of current, voltage, and power in electrical systems. During Russia's war against Ukraine, our energy industry has become very vulnerable to enemy attacks, and therefore needs a quick and effective recovery. Energy specialists lack software tools for working with the power system, and in the period of development of artificial intelligence, creating such tools is not so difficult.

For example, familiar specialists often have to build vector diagrams - tools for visualizing and understanding the distribution of current, voltage and power in electrical systems. Using a combination of frameworks for working with artificial intelligence and creating a graphic shell, you can achieve the desired result in a few months, and at the same time do a useful thing for our victory.

The YOLO model (based on the PyTorch framework) and the QT framework are considered among the proposed tools for creating an intelligent vector diagram construction system. The role of artificial intelligence is to recognize electrical elements in circuits and their connections to each other. Creating a user interface is an equally important thing, which is already implemented in many places with the help of QT.

As of today, there is no specialized software tool for solving the problems of manual construction of vector diagrams, but the proposed approaches are already used for its creation.

The system uses the Yolov5 model to recognize electrical elements on the circuit. The model is trained on more than 150 images and is able to recognize hand-drawn diagrams. Recognition is run as a separate process from the main program written in C++. This part of the system processes the input data from Yolo, saves it in a convenient format, creates a user interface, and displays the result as a vector diagram.

Key words: vector diagrams; framework; power system; artificial intelligence; image recognition; Yolo.