# Project Proposal: Solving advanced puzzles with AI methods

## Project Title

Image Recognition and Solver for Nonogram, Kakuro, and Sudoku

## Introduction

Puzzles such as **nonogram**, **kakuro**, and **sudoku** are popular logic-based games that engage millions of enthusiasts globally. However, solving these puzzles can be time-consuming and challenging, especially for complex grids. This project aims to develop an AI-powered mobile application that uses image recognition to scan, interpret, and solve these puzzles accurately. By leveraging computer vision and advanced algorithms, the app will provide instant solutions.

## Objectives

1. **Image Detection**: Develop a robust image recognition system to detect and digitize puzzle grids from printed or handwritten sources.
2. **Differentiate Image:** Differentiate between each puzzle whether it is a nonogram, kakuro or sudoku puzzle
3. **Solution Generation**: Implement efficient algorithms to solve nonograms, kakuro, and sudoku puzzles.
4. **Accessibility**: Create a user-friendly interface that supports mobile applications

## **Scope**

The project will focus on three puzzle types:

1. **Nonogram**: Interpret black-and-white pixel patterns 5x5 based on given numeric clues.
2. **Kakuro**: Solve logic-based crossword-style 5x5 puzzles involving sums.
3. **Sudoku**: Process 9x9 grids (and potentially larger variants) to identify and fill missing numbers.

## Technologies and Tools

1. **Programming Language**: Python (for algorithms), Kotlin (for Android app development).
2. **Frameworks**: TensorFlow or OpenCV for image recognition, NumPy for matrix operations.
3. **Development Tools**: Android Studio, Jupyter Notebook (for prototyping).
4. **Databases**: Firebase or SQLite for storing puzzle data.

## Motivation and Background

Puzzle-solving has been a popular mental exercise for centuries, evolving from traditional pen-and-paper formats to digital platforms. Advanced puzzles like Sudoku, Nonogram, and Kakuro are not just recreational activities but tools for cognitive development, enhancing logical thinking, pattern recognition, and problem-solving skills. Despite their growing popularity, solving these puzzles can be time-consuming and challenging, especially for beginners or when tackling complex variations.

The advent of artificial intelligence (AI) and image processing technologies presents an opportunity to bridge this gap, enabling automatic recognition and solving of puzzles. Current solutions often cater to individual puzzles alone (not a variety of them), lack accuracy in real-world recognition (e.g., scanning puzzles from newspapers or books)

This project is motivated by the desire to create a comprehensive app that combines AI and image recognition to accurately interpret and solve puzzles from various sources. The app will enhance accessibility users can scan puzzles from printed materials or screenshots, eliminating manual input. Also saves time complex puzzles can be solved instantly, catering to users who enjoy puzzles but are constrained by time.

The main issue is there is no app that caters towards solving a wide range of puzzles only individual puzzles. With this project I want to break that barrier and open a door to a unified solution that caters to multiple advanced puzzles within a single platform.

## Project Plan

A Identify and evaluate existing puzzle-solving apps

B Research image recognition techniques and frameworks

C Interview/issue questionnaire to puzzle enthusiasts

D Write supervisor project proposal.

E Source sample puzzles (Nonogram, Kakuro, Sudoku)

F Identify suitable image preprocessing algorithms

G Install and configure image processing libraries (e.g., OpenCV)

H Implement grid detection for puzzles

I Configure OCR for digit and character recognition

J Write interim report for dissertation (deadline 30 December).

K Develop algorithms for solving Sudoku puzzles

L Develop algorithms for solving Kakuro puzzles

M Develop algorithms for solving Nonogram puzzles

N Integrate algorithms with image recognition system

O Test puzzle recognition and solution accuracy

P Design user interface for scanning and viewing puzzles

Q Develop step-by-step solution explanation feature

R Conduct usability testing with sample users

S Perform final system testing

T Collect user feedback from beta testing

U Evaluate app against objectives

V Finish presentation

X Finish final project report

