پروژه: سیستم هوشمند تشخیص و حل سودوکو

مبانی بینایی کامپیوتر

استاد درس: دکتر محمدرضا محمدی

دستیار: مهرشاد فلاح

مهلت تحویل: ۱۴۰۴/۰۴/۱۹

۱) مقدمه و هدف پروژه

هدف این پروژه، طراحی یک سیستم بینایی کامپیوتر برای تشخیص و حل خودکار سودوکو از روی تصویر است. سیستم باید بتواند جدول سودوکو را از تصویر استخراج کرده، ارقام را تشخیص دهد و پاسخ نهایی را نمایش دهد.
مباحث اصلی پروژه شامل:

- پردازش تصویر با OpenCV
- شبکه‌های کانولوشنی (CNN)
- تشخیص ارقام
- طراحی یک Pipeline کامل بینایی کامپیوتر

تمرکز اصلی پروژه بر بخش بینایی کامپیوتر، استخراج صحیح جدول و تشخیص ارقام است. برای بخش حل سودوکو، استفاده از پیاده‌سازی‌های استاندارد الگوریتم Backtracking مجاز است. دانشجویان می‌توانند از پیاده‌سازی‌های آموزشی و متن‌باز موجود استفاده کنند.

۲) دیتاست و منابع داده

- استفاده از دیتاست‌های استاندارد مانند:
  - MNIST
  - Chars74K
- استفاده از دیتاست ارقام فارسی مجاز است. نمونه مناسب:
  - Hoda Dataset
- تولید داده مصنوعی با استفاده از:
  - تغییر فونت
  - چرخش
  - نویز
  - تغییر روشنایی
- استفاده از تصاویر واقعی سودوکو برای تست نهایی الزامی است.

۳) فازهای پروژه

فاز ۱ – استخراج جدول سودوکو

- تبدیل تصویر به Grayscale
- حذف نویز و استخراج لبه‌ها
- تشخیص جدول با:
  - findContours
  - یا Hough Transform
- اعمال Perspective Transform
- تقسیم جدول به ۸۱ سلول
- حذف خطوط جدول و تشخیص خانه‌های خالی
- **تحویل:**
  - تصاویر مراحل پردازش
  - نمونه خروجی سلول‌ها
  - تحلیل نمونه‌های شکست

فاز ۲ – تشخیص ارقام

- طراحی و آموزش مدل تشخیص رقم
- ارقام می‌توانند:
  - فارسی
  - انگلیسی

  باشند.
- تعریف مدل به‌صورت ۱۰ کلاسه:
  - اعداد ۱ تا ۹
  - کلاس خانه خالی
- استفاده از:
  - CNN
  - یا معماری‌های سبک مانند MobileNet

- استفاده از:
  - Data Augmentation
  - Normalization
  - CrossEntropy Loss
- ارزیابی با:
  - Accuracy
  - Confusion Matrix
- **تحویل:**
  - نمودار آموزش
  - نتایج ارزیابی
  - تحلیل خطا

فاز ۳ – حل سودوکو

- تبدیل خروجی مدل به ماتریس ۹×۹
- نمایش خانه‌های خالی با مقدار صفر
- استفاده از الگوریتم Backtracking
- بررسی اعتبار اعداد در:
  - سطر
  - ستون
  - بلوک ۳×۳
- مدیریت جدول‌های ناسازگار یا بدون پاسخ
- استفاده از پیاده‌سازی‌های آموزشی و متن‌باز برای Solver مجاز است.
- نمونه منابع پیشنهادی:
  - https://github.com/LiorSinai/SudokuSolver-Python
  - https://github.com/aimacode/aima-python

فاز ۴ – سیستم نهایی

- اتصال کامل مراحل پروژه:
  - استخراج جدول

  - تشخیص ارقام
  - حل سودوکو
- نمایش پاسخ نهایی روی تصویر اصلی
- تحلیل عملکرد سیستم:
  - دقت
  - زمان اجرا
  - تحلیل خطا
- تهیه گزارش نهایی

۴) استاندارد تحویل

پروژه باید قابل اجرا و قابل بازتولید باشد.

- ارائه فایل requirements.txt یا environment.yml
- ارائه وزن مدل نهایی
- ارائه دستور اجرای پروژه
- تمامی بخش‌های پروژه باید به‌صورت فایل‌های .py تحویل داده شوند.
- تحویل صرف پروژه در قالب Jupyter Notebook قابل قبول نیست.

۵) ارزیابی پروژه

- کیفیت استخراج جدول سودوکو
- دقت تشخیص ارقام
- عملکرد صحیح سیستم نهایی
- تحلیل خطا و نمونه‌های شکست
- کیفیت گزارش و ساختار پروژه

۶) بخش امتیازی

گزینه ۱ — مقاوم‌سازی سیستم در شرایط واقعی

- عملکرد صحیح سیستم در شرایط دشوار تصویری مانند:
  - نور کم
  - نویز
  - تاری تصویر
  - چرخش
  - زاویه دید شدید
  - سایه
- تحلیل عملکرد سیستم در این شرایط
- استفاده از تکنیک‌های مناسب برای افزایش Robustness

گزینه ۲ – بهینه‌سازی و استقرار مدل

- تبدیل مدل به:
  - ONNX
  - TorchScript
- تحلیل سرعت اجرا روی CPU
- مقایسه زمان اجرا و حجم مدل قبل و بعد از بهینه‌سازی

گزینه ۳ – رابط کاربری (UI)

- طراحی رابط کاربری ساده برای اجرای سیستم
- نمایش:
  - تصویر ورودی
  - جدول استخراج‌شده
  - پاسخ نهایی
- ابزارهای پیشنهادی:
  - Streamlit
  - Gradio
  - PyQt

گزینه ۴ – نمایش پاسخ روی تصویر اصلی

- نمایش پاسخ نهایی سودوکو روی تصویر ورودی

- حفظ پرسپکتیو اولیه جدول
- هماهنگی صحیح متن و تصویر خروجی

۷) قوانین پروژه

- پروژه به‌صورت دو نفره انجام می‌شود.
- هر دو عضو باید بر تمامی بخش‌ها مسلط باشند.
- استفاده از منابع آموزشی و پروژه‌های متن‌باز مجاز است.
- در جلسه ارائه ممکن است درباره تمامی بخش‌های پروژه سؤال شود. (به غیر از الگوریتم حل سودوکو)