Developer's Network Project 2 (October 2024)

MCR Scottsdale LLC

Welcome to Developer's Network! This packet will serve as the primary source of information on Project 2. This project is designed to engage participants in practical machine learning challenges that will sharpen your problem-solving skills within the agricultural industry, specifically hydroponics.

Objective:

The objective of this project is to get students comfortable with machine learning techniques and their application within the agricultural industry, specifically hydroponics. Each team must consist of 3 to 4 members and select only one task for submission. Multiple submissions from the same team will not be accepted.

Questions and Contact:

For any inquiries or clarifications, please email us at devnet.csu@gmail.com or meet us in person at our weekly meetings every Wednesday at 5:30 PM in CSB130.

Submission Deadline:

All final submissions must be completed and submitted by **November 5th at 11:59 PM**. Presentations will take place on **November 6th at 5:30 PM in CSB130**.

Notice:

Participation in this project requires that all participants and teams abide by any copyright and license agreements on any and all resources used.

Challenge 1: Anomaly Detection

1st - $800, 2nd - $300

Description:

Students will use online datasets to train a convolutional neural network (CNN) to recognize plants as healthy or not. The solution should include a pipeline that processes new image data, sends it to the neural network for classification, and updates a web page with the latest results. Students will also be provided with fake sensor data, which can be displayed on the web page for additional context.

Folder link:
https://drive.google.com/drive/folders/1u8ZzB1cB5ClDCD7g6sNikSLksbGr9j4f?usp=sharing

Example Dataset: https://www.kaggle.com/datasets/csafrit2/plant-leaves-for-image-classification

Requirements:

- Train a neural network (e.g., a simple CNN) using the provided dataset.
- Build a data pipeline that processes new image data and sends it to the model for real-time classification.
- Create a web page that displays the classification results (e.g., “healthy” or “not healthy”) as they are processed, along with a simple visualization of the provided sensor data.
- Evaluate model performance with relevant metrics

Objective: Create a dashboard which takes in an image and sensor data as input and lets a user know if their plants are healthy or not.

Challenge 2: Data Pipeline

1st - $400

Description:

Create an automated data pipeline that integrates with a Google Drive folder. This pipeline should monitor the folder for new data (e.g., sensor readings, images), process it automatically, and update a simple visualization or dashboard to show the latest conditions of a hydroponic system.

Folder link:
https://drive.google.com/drive/folders/1u8ZzB1cB5ClDCD7g6sNikSLksbGr9j4f?usp=sharing

Requirements:

- **Set Up Google Drive API Access**: Use the Google Drive API to monitor a shared folder for new data. The app should have **read-only access**, allowing it to fetch new files when they are added.
- **Automate Data Retrieval**: Create a script or tool that checks for new files in the shared folder and retrieves them.
- **Process Data**: Once new data is retrieved, process it (e.g., format the data, extract relevant information).
- **Update a Dashboard**: Visualize the latest data in a simple dashboard. The dashboard should show current conditions (e.g., temperature, humidity, or any relevant readings).

Objective: Create a tool that takes in image data from google drive and prepares it to be tested on by a trained machine learning model or AI. Display the processed data.

Submission:

Please send a single zip file before the due date to Thomas McRoskey on Microsoft Teams. Include a README.txt file with instructions for running your code, as well as a brief description of your program, especially highlighting any additions not specified in the challenge requirements. Make sure to include link to source code, not just executable files.

Presentation:

One-fifth of each team's total score will be based on their presentation. It's important to note that the presentation score is independent of the technical quality of your program; it focuses solely on how effectively you communicate your approach and solution.

Each presentation will be evaluated based on the following criteria:

1. **Clarity and Structure**
   - How clearly and logically is the presentation organized?
   - Are the objectives, approach, and outcomes of the project clearly stated?
2. **Engagement and Delivery**
   - How well does the team engage the audience?
   - Are the presenters confident, articulate, and engaging?
3. **Visual Aids and Design**
   - Are the slides or visual aids well-designed, clear, and free of unnecessary clutter?
   - Do they enhance the understanding of the project?
4. **Explanation of Approach**
   - How effectively does the team explain their problem-solving approach and methodology?
   - Is the explanation detailed enough to convey the thought process behind key decisions?
5. **Team Collaboration and Communication**
   - Does the presentation reflect balanced participation and collaboration among team members?

Good Luck!

We look forward to seeing your innovative solutions and wish you the best of luck!

Judging Scorecard

Key Attributes		Score
Idea	Did the proposal address the problem statement and the theme?	/10
	Was the idea innovative?	/15
Implementation	User experience	/15
Presentation	Does the presentation clearly define and address the problem statement?	/20
Scalability	Is the implementation easy to add to? Is it well documented?	/10
Total		/70

Resources:

Machine Learning & Neural Networks:

- **TensorFlow**: Build and train neural networks using this popular library.
  - TensorFlow Tutorials
- **PyTorch**: An intuitive alternative to TensorFlow for deep learning.
  - PyTorch Tutorials

Data Handling & API Integration:

- **Google Drive API**: Interact with files and folders in Google Drive programmatically.
  - Google Drive API Documentation
- **OpenCV**: Comprehensive tools for preprocessing data.
  - OpenCV Documentation

Data Visualization & Dashboards:

- **Streamlit**: Create interactive dashboards with minimal code.
  - Streamlit Documentation
- **Dash**: Build web-based dashboards for displaying data.
  - Dash Documentation

Python Libraries for Data Processing:

- **Pandas**: Ideal for reading and analyzing data.
  - Pandas Documentation
- **Matplotlib**: Create charts and visualizations.
  - Matplotlib Documentation