Analyzing Yellow Rust on Wheat

A machine learning approach and analysis

Reiko Uygongco, Sean Aguilar

Identifying Wheat

abstract

- Yellow rust threatens wheat yields globally
- Used “Yellow-Rust-19” Kaggle dataset by Tolga Hayit
- The goal of the project was to detect disease severity with the help of machine learning
- The main thing we found was how an image based model can help Farmer/researchers identify the problem of yellow-rust early.

Intro and problem

- Wheat is a vital crop to the world and yellow rust decreases yields by 20-30%
- The problem lies within having to manually detect each individeal wheat plant
- Objective was to build an autommated system that will classify using the data set
- We hoped to boost precision in agriculture and food security

Literature review

- Researchers have started using smart tech to spot yellow rust in wheat.
- CNNs Work Well: Hayit et al. (2021) used a type of computer brain (CNN) to figure out how bad the rust is
- Drones Help: Su et al. (2018) flew drones with special cameras to watch crops from above.
- Mixing Clues: Guo et al. (2020) combined color and texture details from images, hitting 80–98% accuracy.
- What’s Missing: We need faster, real-world tools to catch rust early and help world of agriculture.

dataset description

- "YELLOW-RUST-19" from Kaggle (Tolga Hayit).
- Likely images of wheat leaves (healthy vs. infected).
  - There were two image folders, Raw classified images and the processed images
    - The raw images are grouped based on the plants overall health while the processed are grouped based on the leafs disease severity
- 2000+ images, labeled by severity.
- Split: Training + testing sets.

explaratory data analysis

- Class split : Even amount of images in each category ranging from healthy, mild, moderate, severe
- Image check: Resolution, color, rust patterns.
- Texture: Infected leaves show more variability.


O


R


MR

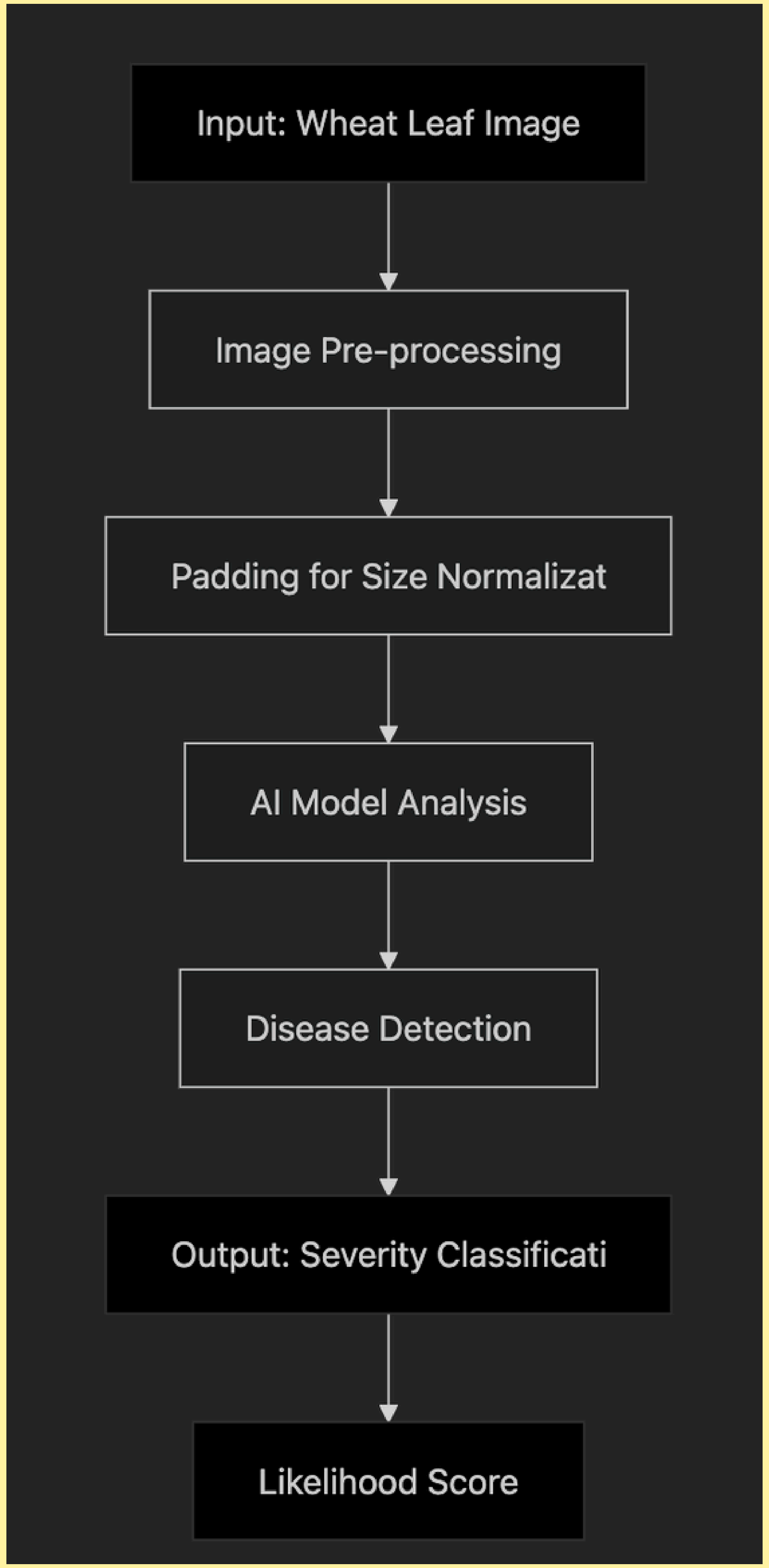

MRMS


MS


S

visualization

- The images from the data set were taken and we used a code to pad the images to account for the size differences present in the original image set
- These padded images were then used to train the AI model to identify two categories, Diseased and Healthy.



model developement and evaluation

- Initial Approach
  - Used Google Teachable Machine for training and disease identification
  - Encountered issues with varying image sizes, implemented padding solution
  - Found discrepancies between Teachable Machine and Copilot performance
- Iteration and Improvements
  - Attempted Python-based model but faced category similarity issues
  - Switched from regression to binary classification model
  - Successfully achieved accurate image categorization using Teachable Machine
- Technical Details
  - Originally designed for 6 severity classifications
  - Simplified to binary classification (Resistant vs. Severe)
  - Reduced dataset from 10,000+ to 4,000 images to improve performance
  - Image padding improved accuracy from 50% to 86%
- Evaluation
  - this is a good model in theory but for it to be great, it would be better if the model was able to identify the images regressionaly instead of binarialy
  - There are many possibilities with this model but that would be for a more advanced level of coding and modelling

Results

- After the regression code ran into issues due to the amount of similarities between the plant disease severity as they looked very similar to each other
- This led us to doing severities binarily and picking the two on opposite ends of the scale (R) and (S)
- Finally with this code we were able to use google teachable machine and train the model to identify the diseases.
- At first it was identifying the images with a 50% accuracy which is the same as a blind guess and can be compared to flipping a coin.
    - but with more tweaks we were finally able to with around an 86% or more accuracy which was better than the initial 50% which with 2 categories is basically a blind guess.
        - It was found after troubleshooting that it was the padding of images that increased the accuracy of identification
- After all the troubleshooting we were finally able to run the code and identify leaf disease, however it is important to note that due to this being a binary identification model it is more likely that some sign of disease will show up as sever -> based on testing the model

Conclusion

- Model Performance
  - Successfully developed initial leaf disease classification system
  - Dataset proved valuable for model training
  - It is a good stepping stone and starter/basic model
- Current Challenges
  - Images must be standardized to same size for optimal accuracy
  - Model struggles with distinguishing between similar pictures thus made it harder to create a regression model as we kept running into challenges
- Future Development Opportunities
  - Potential conversion into farmer-friendly mobile app that can be used to take a picture of a leaf and determine if it is diseased and the severity of the disease (assuming that a regression model was able to be made)
  - Disease severity assessment capabilities
  - Integration with security cameras for automated field monitoring
    - Reduced need for manual field inspection