Deep Handwriting Recognition Model Presentation

William Farris, Baron Davis, Micheal Oyenekan, Creighton Young

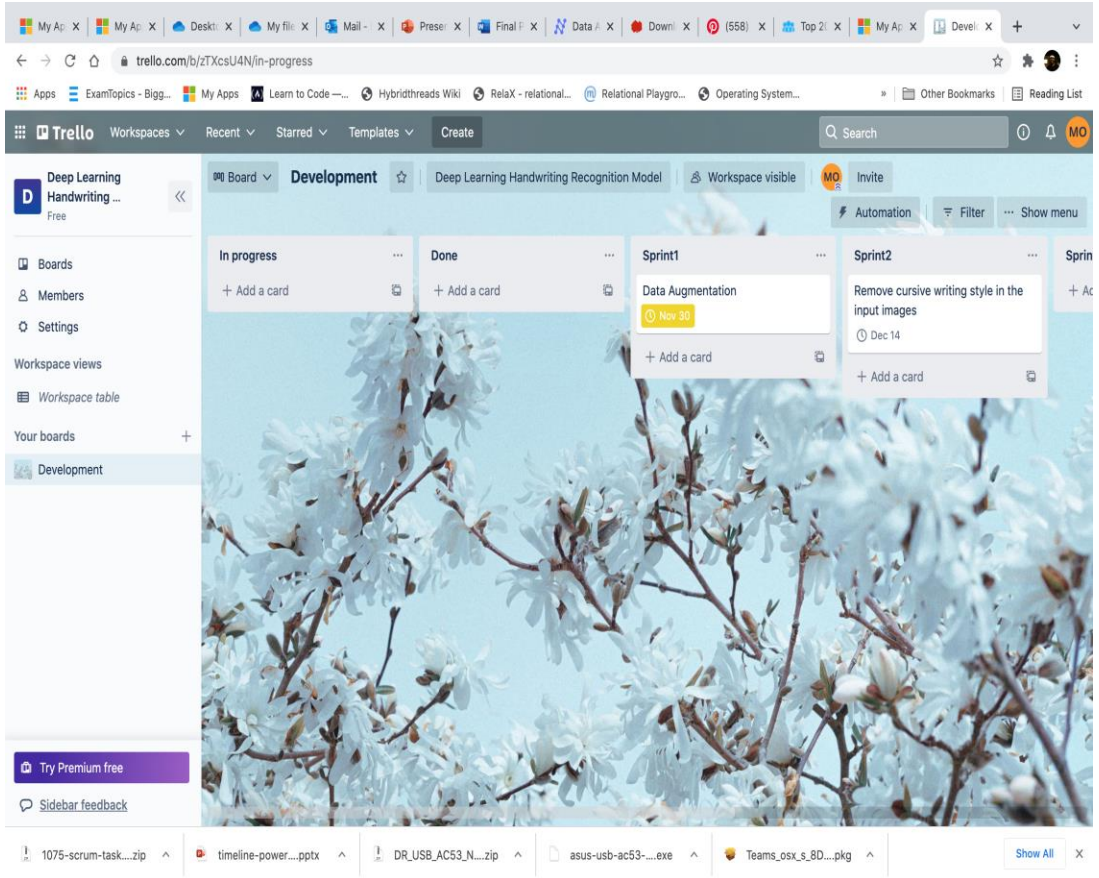
Abstract - Problem & Objective

- Current problems:
  - World becoming more and more digital
  - Sometimes things must be hand recorded as bringing devices can be inconvenient.
  - Translating written information to being machine readable can take time.
- Objective:
  - Build upon an already existing Handwritten Text Recognition model.
  - Raise the current 70% to at least 90%.

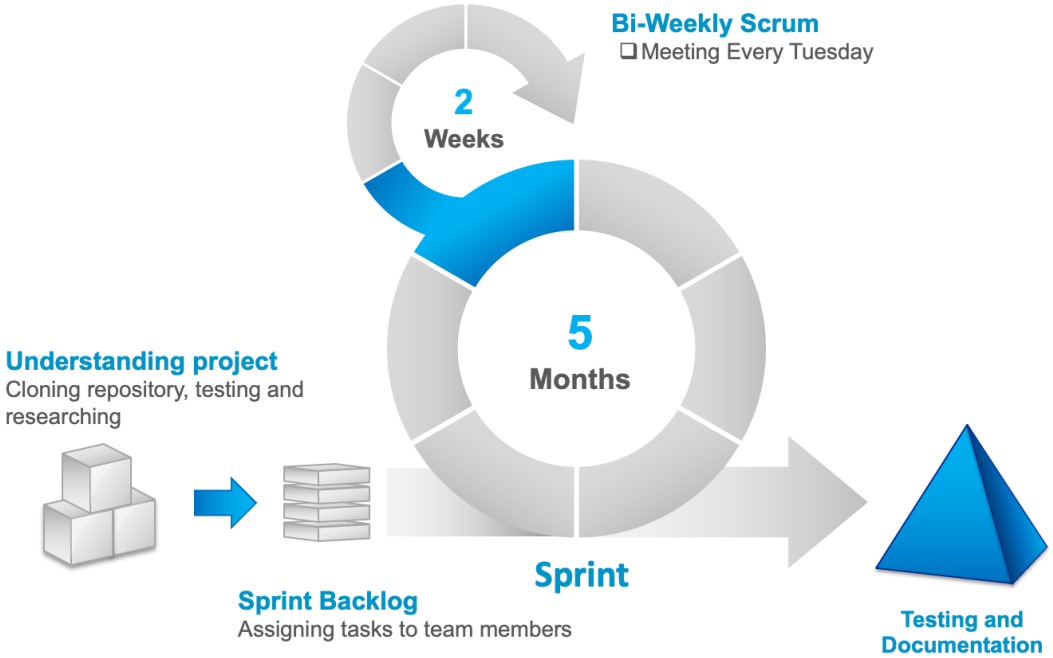
Background – Key Concepts

- Convolutional Neural Network (CNN)
  - Takes an input image and assigns importance into different layers
  - An Adam Optimizer allows us to update importance, learning rate, and reduce error
  - RELU Function – Determines relevance in a CNN layer
- Recurrent Neural Network (RNN)
  - Recognize and remember time and sequences of data inputs
  - Long Short-Term Memory (LSTM & 2D LSTM)
- Handwriting can vary
  - Connectionist temporal classification (CTC) – Allows us to determine relevancy when handwriting size varies
  - Word Beam Search and Token Passing – Algorithms that allow us to decode similar looking letters

Design – Tasks



Deep Learning- Handwriting Recognition Scrum process



Design – Schedule

TASKS	DATES
1. Meeting with Industry champion on what needs to be done…	10/19
2. Get familiar with repo, AI, and python	10/26-11/9
3. Data augmentation	11/16 - 11/30
4. Remove cursive writing style in the input image	11/30 - 12/14
5. Increase input size	1/11 - 1/25
6. Add more CNN layers	1/25 - 2/1
7. Replace LSTM by 2D-LSTM	2/1 - 2/15
8. Replace optimize	2/15 - 3/1
9. Decoder	3/1 - 3/15
10. Text correction	3/15 - 3/29
11. Testing	3/29 - 4/19
12. Documentation	4/19- 5/3

Design - Key Components

- Deliverables
  - Design Document
  - Initial Data
  - C++ code
  - Final Report
- Personnel
  - William Farris, Baron Davis, Micheal Oyenekan, Avery Lang, Creighton Young
  - Nathaniel Zinda
- Key Facilities and Equipment
  - Personal computers