CS 5232 - Written Project Proposal

Eric Nieters, Ibrahim Mamudu, Raj Saxena

Objectives:

- Identify the first six generations of Pokémon with image recognition.
- Detect more than one different "style" of Pokémon.
    - Artwork, TV shows, card portraits, video games.
- Provide more information than just the name.
    - ID index, gender, height, weight, and more.

Contributions:

- Provide full information regarding the Pokémon through image recognition.
- Create a real-life "Pokédex" by using your camera or a photo.
- Create a training tool for players who want to learn about the hundreds of Pokémon that were in the first six generations.
- Identify not only based on original artwork, but also creations that artists have made.
- View other types of Pokémon that occur in the same evolution.

Machine Learning Algorithms:

- Convolutional Neural Networks (CNNs) - This will be our "from scratch" algorithm.
- Transfer Learning with Pre-trained Models
- Support Vector Machine
- K-Nearest Neighbor

Related Work:

- Kadlaskar, A. (2021, June 14). *Image Classification Using Convolutional Neural Network with Python*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/image-classification-using-convolutional-neural-network-with-python/
    - This article will help provide an example of using CNN with Python, giving us a general direction of where we need to go with how to begin our project. This reference gives a thorough example, which we will be working off of.
- *Mwiti, D. (2023, August 22). Transfer learning guide: A practical tutorial with examples for images and text in Keras. neptune.ai. https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras*
    - This is a very recent guide that provides an example of how transfer-learning will work in Python. It also provides a very beneficial example which uses images for classification, which we will need to look at for our project.
- Gautam, T. (2023, July 31). Create your own image classification model using Python and keras. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/create-image-classification-model-python-keras/
    - This source is a mix of the two sources above, which talks about the CNN model using Python, as well as using Keras, which was spoken about in the second reference. This article also discusses how to distinguish between two very close subjects (rugby and soccer), which we will have to put into consideration because many Pokèmon appear the same as one another within the evolution line.

- Rosebrock, A. (2021, April 20). Your first image classifier: Using K-NN to classify images. PyImageSearch. https://pyimagesearch.com/2021/04/17/your-first-image-classifier-using-k-nn-to-classify-images/
    - This last source is an incredibly useful resource which we will use if we struggle with the CNN algorithm. This source uses the K-Nearest Neighbor algorithm, with an elementary introduction that discusses how the algorithm works. It will also use libraries that we plan to use with the CNN approach, such as NumPy and OS.

Research Methodology and Timetable:

Task #	Description	Deadline	Who?
1	Prepare for a work environment and get GitHub and Python prepared for work.	9/26 - 10/2	Everyone
2	Review Python syntax, functions, and determine the libraries that need to be used for the project.	9/26 - 10/9	Ibrahim/Raj
3	Prepare a dataset for all Pokémon images.	9/26 - 10/9	Eric
4	Prepare an image recognition program that can detect a small amount of Pokémon (roughly 10-20). Determine if other algorithms may be required. - Use CNN first to determine if this is suitable for the project's needs. CNN will be implemented from scratch. This will take the most time. - If the program can use CNN, but doesn't provide efficient results, we will use a different algorithm. - Implement the K-NN approach. - Consider pre-trained models or support vector machines if time allows us to.	10/9 - 10/31	Eric - Writing CNN algorithm Raj - Writing K-NN algorithm Ibrahim - Writing any algorithm that's left.
5	Have a progress report written and submitted by this date.	10/22 - 10/24	Everyone

6	Repeat Task 4, but with the other two algorithms, as well. After completion, test the data with all of the Pokémon that's been gathered from earlier. All 721 Pokémon. Begin to add all the data that correlates with the Pokémon that has been detected (stats, height, weight, gender, etc).	10/31 - 11/14	Everyone (will separate across the different work that needs to be done).
7	Complete the preliminary writeup.	11/14 - 11/16	Everyone
8	Fine-tune anything else that needs to be done. Add additional features that could make the project appear better, like the app implementation or a website.	11/16 - 12/1	Everyone
9	Complete final oral presentation/report.	12/1 - 12/7	Everyone

Addressed Comments:

- *"Has work like this on Pokémon been done before? If so, who did it? What are the differences with what you are proposing?"*
    - There have been a couple Pokémon examples in the past, but we plan to implement more than just providing the names of the Pokémon (like what a couple tutorials have done already). Since we have a dataset of Pokémon, as well as their stats, we will be adding some more information to the results of our program. To make sure that we get the most out of this project, we will not be following tutorials that are specific to identifying Pokémon.
- *"This type of project can be made even more useful if you attempt to make a phone app to recognize pokemon. It would make your app a legitimate pokedex. Are you considering this idea?"*

  - We might consider this idea if we have enough time. However, not all of our group members have worked in Python before, so turning this into an app might require more time than we'd expect. However, if we make enough progress, we will find ways to improve upon it and will consider making an app out of it.
- *"What are the four algorithms you are planning to use?"*
  - We forgot to mention two more in our slides, but we included an additional two algorithms (for a total of four) on the first page.
- *"Which algorithms are you implementing by hand from scratch?"*
  - We will likely be implementing the CNN algorithm by scratch. We would have to stay away from using the pre-trained model algorithm by scratch because that would require a model already developed and re-worked to fit with our project. We will also consider working on the K-NN algorithm, provided we can find examples of how the algorithm works. We have already found one (resource 4), so we will likely be building this instead of CNN if CNN doesn't work for us.