Interacting with Machine Learning

AnimaDex

Sandra HATEM, Martin GADET, Piotr WAWRZYNIAK

1. Introduction

AnimaDex is an interactive machine learning project designed to classify animals using a mobile-friendly web application. Inspired by the concept of a Pokédex, AnimaDex allows users to upload images of animals, contribute to dataset expansion, and improve classification accuracy through model retraining. The system utilizes a combination of MobileNets as a feature extractor and K-Nearest Neighbors (KNN) for classification, making it lightweight and efficient.

2. Concept and Design

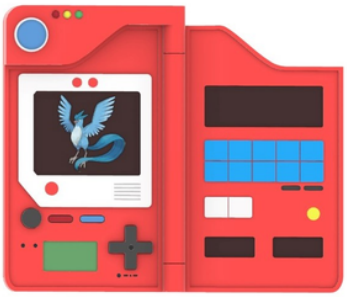
The primary goal of AnimaDex is to provide an engaging and educational experience for animal lovers, explorers, and gamers. The application functions on both mobile devices and web browsers, ensuring accessibility in various environments such as homes, zoos, or the outdoors.

User Goals:

- Capture and classify images of animals.
- Contribute to dataset expansion by uploading new images.
- Improve model accuracy through interactive retraining.
- Explore different animal species via an intuitive interface.

Key Features:

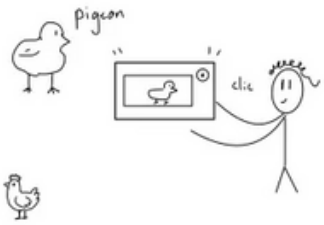
- Dataset Upload: Users can upload image datasets in bulk by selecting entire folders.
- Model Training: A simple interface allows users to train and retrain the classifier.
- Real-time Classification: Users can classify images and provide feedback to enhance accuracy.
- User Feedback Mechanism: Images can be added to the dataset based on user confirmation.



PokeDex

Pokemon Go

Sketches:



User take picture

Model do a prediction
User can add or not

User explore the dataset

3. Implementation

The dataset is managed using Marcelle.js and stored in-memory for rapid processing. The initial dataset consists of 90 animal classes with 5,400 images. Users can extend this dataset by uploading new images, which are automatically labeled based on their folder structure.

Feature Extraction and Classification

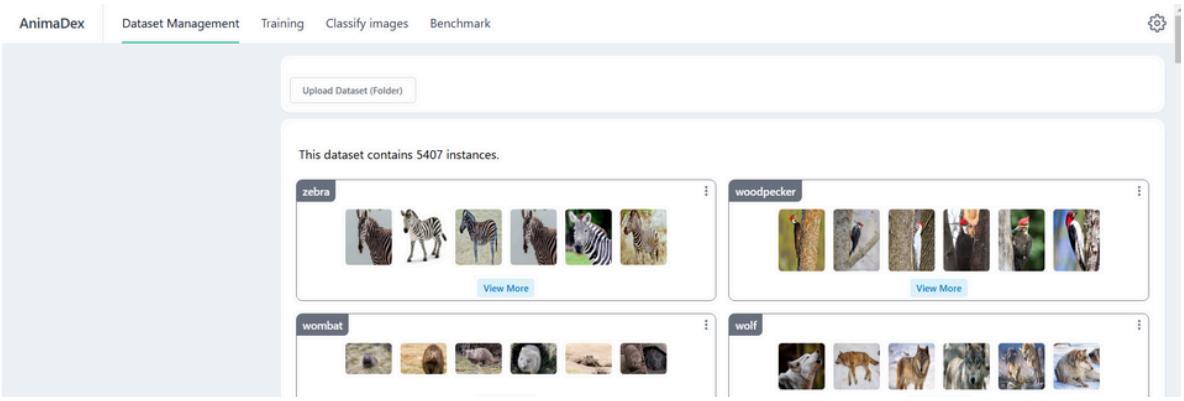
- Feature Extraction: MobileNets is used to generate feature embeddings from images.
- Classification Algorithm: A KNN classifier, synchronized with the data store, is employed for efficient and lightweight classification.
- Training Pipeline: Images are pre-processed, and features are extracted before training the KNN model.

Interactive Classification Process

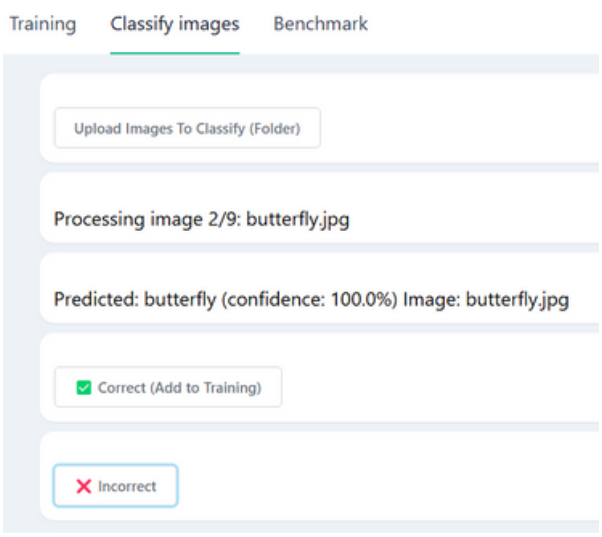
1. Users upload new images to classify.
2. The system predicts the class using the KNN model.
3. Users validate predictions by marking them as correct or incorrect.
4. Correct classifications are added to the dataset, allowing incremental model retraining.
5. Accuracy is continuously monitored and displayed to users.

Training dataset:

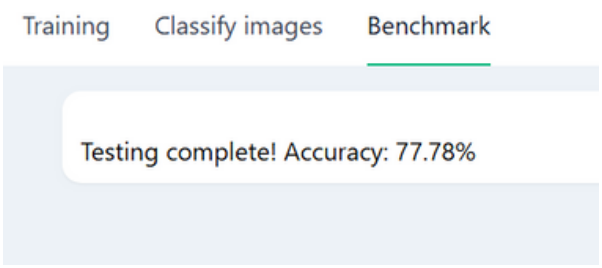
Screenshots of application:



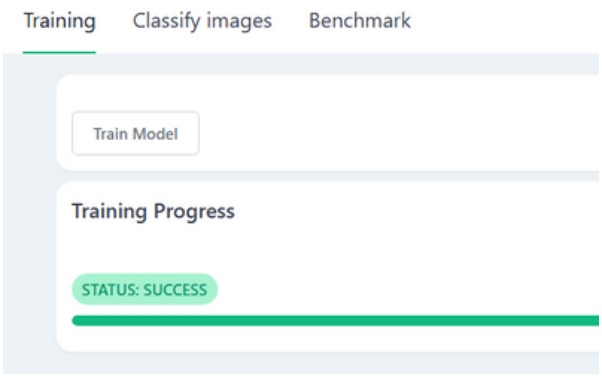
Dataset panel



The user can upload folder of images and add or not the images one by one based on the model prediction and accuracy



The user can see the global accuracy of the model that he can increase over time



The user can retrain the model with new images

4. Evaluation and Results

Model Performance Testing

- A split-test evaluation compared the model's accuracy before and after retraining.
- Initial accuracy was measured on a fixed test set.

User Experience Testing

- Usability Testing: Observations and feedback were collected from users interacting with the interface.
- Engagement Metrics: Number of images uploaded, retraining frequency and frequency of use
- Hypothetical Application Testing: Although users did not have direct access to test the application, a theoretical scenario suggests that if they had, they would have been able to upload their own images and content for testing purposes. This would have allowed for evaluation of how well the system adapted to real-world data provided by different individuals and how effectively users could enhance the dataset through interactive retraining.

5. Conclusion

AnimaDex successfully demonstrates an interactive approach to machine learning, allowing users to actively participate in dataset expansion and model refinement. The combination of MobileNets and KNN ensures efficiency, making it suitable for real-time classification on mobile and web platforms. Future enhancements could include deep learning-based classification, cloud-based storage for larger datasets, and integration with augmented reality (AR) for an even more immersive experience.