Andrii Denysenko

(looking for remote work)

andriiy.denysenko@gmail.com • (+421) 952 027 754 / (+380) 67 648 2446 • Kosice, Slovakia

SKILLS

Programming languages/tools:

- 4 years of experience in software development
- Languages: Proficient with Python, experienced in C and C#, experienced in HTML and CSS
- AI Libraries: PyTorch (built AI models based on Iris), TensorFlow, Keras, Ultralytics, Yolo and scikit-learn
- Tools & Technologies: Git, Jupyter Notebook, Google Colab, Matplotlib, Pandas, NumPy, Tkinter, Telebot

EDUCATION

“Prestige” High School with Math and Programming Specialization - Kyiv, Ukraine	Sep 2020 - Jun 2023
Technical University of Kosice - Bachelor in Computer Science - Kosice, Slovakia	Sep 2023 - now

PROJECTS

September-December 2024

Neural-network app to classify dog breeds using Keras Tensorflow(AI/ML group project)

- Built a convolutional neural network for dog breed classification using the Stanford Dog Breed dataset, experimenting with advanced architectures like InceptionV3, ResNet50, and ResNet101 to optimize performance.
- Enhanced model training by evaluating and fine-tuning optimizers such as Adam, SGD, and RMSprop, achieving improved accuracy and convergence rates.
- Applied data augmentation techniques using Keras and TensorFlow to increase dataset diversity, leading to better model generalization and reduced overfitting.
- Monitored and analyzed training metrics using Pandas, enabling informed adjustments to preprocessing, hyperparameters, and learning strategies.

Telegram bot that uses neural-network for dog breed classification(AI/ML group project)

- **Model Development:** Train a CNN using the Stanford Dog Breed dataset with architectures like ResNet50/InceptionV3, apply data augmentation, optimize with Adam/SGD, and export the model in `.h5` or SavedModel format.
- **Bot Functionality:** Use `Telebot` to enable image uploads, preprocess images, predict the breed using the trained model, and send results (breed name, confidence score) to users.

- **Backend and Deployment:** Host the bot backend (FastAPI/Flask) on a cloud server or Heroku, integrate the model for predictions, and ensure secure and scalable deployment with Docker if needed.
- **User Interaction and Feedback:** Implement commands like `/start` and `/help`, handle errors gracefully, and allow feedback on predictions to improve the model.
- **Monitoring and Updates:** Track usage metrics, analyze user feedback, and update the model periodically with improved training based on new data.

Website about Artificial Intelligence(Web-development project)

- Developed a blog about AI using HTML and CSS focusing on color scheme and functionality to hold users attention.

July 2024

Developed custom neural-network model using YOLOv8 for Object detection(AI/ML personal project)

- Implemented object detection and classification tasks using YOLOv8, achieving high accuracy and real-time performance.
- Trained the model on custom datasets by leveraging data preprocessing techniques and annotation tools such as LabelImg to ensure high-quality input.
- Experimented with transfer learning by fine-tuning YOLOv8 pre-trained weights on specific datasets to enhance detection precision.
- Applied various data augmentation methods, including random cropping, flipping, and color adjustments, to improve model generalization.
- Optimized training by evaluating different hyperparameters like learning rate, batch size, and confidence thresholds, improving overall performance.

June 2024

Telegram bot for weather forecast using Telebot library and different API(work with APIs and requests, personal project)

- Developed a Telegram bot for weather forecasting using the Telebot library, offering users real-time weather updates and forecasts.
- Integrated multiple weather APIs, such as OpenWeatherMap and WeatherStack, to provide detailed information on temperature, humidity, wind conditions, and more.
- Utilized Python's requests library to handle API requests and responses, ensuring smooth communication and accurate data retrieval.

September-December 2023

Hangman Game (C Programming Project)

- Developed a console-based Hangman game with features to track guessed letters and provide feedback on valid/invalid guesses.
- Implemented logic to dynamically check word progress and ensure user-friendly interaction, including letter availability prompts.
- Enhanced the experience by reading random words from a file for gameplay, improving replayability.

Ball Sort Puzzle Game (C Programming Project)

- Created a game that simulates sorting colored balls into designated columns, focusing on problem-solving and logic.
- Designed a generator function to populate game fields randomly while maintaining valid constraints.
- Implemented user interaction mechanisms for moving balls and checking game completion, providing clear feedback on invalid moves.

Snake Game with Multiple Levels (C Programming Project)

- Designed and implemented a Snake game with features such as multiple levels, increasing difficulty, and adjustable speeds.
- Utilized `ncurses` library to create a dynamic game interface with real-time movement and collision detection.
- Added functionality for saving and displaying the highest score, enhancing user engagement.
- Included advanced level designs with obstacles and game modes for increased challenge and replayability.

LANGUAGES: ENGLISH, RUSSIAN, UKRAINIAN, SLOVAK, GERMAN
