

Muhammad Naeem

✉ m.naeem4288@gmail.com in linkedin.com/in/muhammadnaeem27 🔗 muhammadnaeem27.github.io

Education

Comsats University Islamabad, Pakistan

September 2019 – June 2023

Bachelor of Science in Computer Science

Relevant Coursework: Data Structures and Algorithm, Digital Image Processing, Machine Learning, Computer Vision, Pattern Recognition

Skills

Python, C++, HTML/CSS, NumPy, Pandas, OpenCV, PyTorch, TensorFlow, Keras, Git, AWS, Fast API

Experience

NineSol Technologies

April 2023 – Present

AI Developer

- Developed and integrated the latest models into the backend of mobile and web applications, enhancing functionality with features such as Background Removal, Colorization, Document Scanning, Virtual Try-on, and Transcription as highlighted in my portfolio.
- Stayed informed with the latest industry trends and emerging technologies, applying this knowledge to refine and advance features within diverse applications.
- Collaborated with other team members to smoothly embed advanced functionalities, ensuring seamless user experiences in mobile applications.
- Tools: Python, PyTorch, TensorFlow, YOLO-v5, OpenCV, Keras, Fast API, AWS, GCP

Projects

Weed Detection Robotic Car

Tech Stack: YOLO-v5, TensorFlow, Flask, Tkinter, VS code, Colab, Roboflow, labeling

- Conducted interviews with farmers at the beginning of this project to compile a diverse dataset comprising four categories of weed. Implemented YOLO-v5 for real-time object detection, achieving precise identification of weeds in agricultural fields.
- Designed and built a robotic car prototype that uses a smartphone camera and DC motors to wirelessly stream live images to the YOLO-v5 model via FastAPI. demonstrated a cutting-edge method of precision farming by integrating an Arduino-controlled LED system with a matrix-based system to precisely locate and eliminate weeds that were detected.
- Created a website to showcase project insights, allowing users to engage with our trained YOLO-v5 model and evaluate its effectiveness. Data visualization was used, highlighting the project's technological innovation and potential for sustainable agricultural practices. This included overall and daily weed detection percentages.

Portrait Background Remover

Tech Stack: Python, PyTorch, U-Net, NumPy, Scikit-Learn, OpenCV

- Using *U-Net* structures for both rough and detailed portrait segmentation, an intelligent two-segment Portrait Matting Model was used to address the problem of subject isolation in images.
- The model's resolution limitations (below 2000x2000) and possible biases in the small amount of training data were acknowledged. Used the internet and publicly available datasets like COCO to find examples of portrait photos.

Colorize Gray scale Image

Tech Stack: Python, PyTorch, U-Net, Fast API, Colab, VS code

- Included two time-scale update rules, self-attention *GANs*, and inflection points for effective and error-free coloring.
- The *U-Net* architecture, which includes model-specific backbone choices (resnet34/resnet101), was utilized. It included spectral normalization and self-attention.
- For realistic colorization free of artefacts, Perceptual Loss based on *VGG16* was implemented during NoGAN learning.

Teacher Assistant

Tech Stack: Python, Azure Speech Studio, GPT, PyAudio, Pyannote-Audio, Customtkinter, Firebase database, Firebase Authentication

- Developed a transcription program utilizing *PyAudio* and *Pyannote-Audio* packages, ensuring precise speaker segmentation during the transcription process for both live audio streams and pre-recorded .wav files.
- Used *Customtkinter* python package to create a simple educational interface that made it easier to add subjects, view lecture reports, and create assignments and quizzes based on lectures.
- Firebase's secure cloud functionality was integrated, guaranteeing data security, and enabling features like email lecture report distribution to students for improved accessibility.

Image to Talking Portrait

Tech Stack: Python, Dlib, face-recognition, PyTorch, ffmpeg, Dlib, Open-CV, Fast API, Colab, VS code

- Created talking videos by smoothly combining the lips and head movements of a reference video with a source image. *Dlib* was put into use for accurate facial landmark identification and expression simulation.
- Used *Wav2Lip-hq* for increased realism to ensure precise lip movement synchronization between the source image and the reference video. *ESRGAN* was integrated to up-sample the output video, which improved the visual quality.
- Designed a pipeline that integrates synchronized audio from the reference video to create finalized video. Used *ffmpeg* and *OpenCV* to process and compile videos efficiently.

Document Scanner

Tech Stack: Python, OpenCV, PyTorch, Fast API, Colab, VS code

- Led the creation of a Document Scanner with a Geometric Unwarping Transformer, using advanced algorithms from the Doc3D dataset for accurate geometric unwrapping of document images.
- Using DocProj dataset, the Illumination Correction Transformer was able to resolve illumination problems during document scanning, which improved the quality of the images.

Sky Changer

Tech Stack: Python, PyTorch, UNET, Fast API, Colab, VS code

- Sky Changer model consists of three main components: sky-changing (Multiband blender), high-res processing (UNET), and low-res processing (hrnet-ocr).
- Enhanced segmentation in the low-res module by using ASPP. The learnable parameter high-resolution module performed better than the direct up sampling method. Using a Multiband blender, the sky-changing module effectively replaced the skies.

Text-to Image Generator

Tech Stack: Python, PyTorch, Hugging Face, Stable Diffusion, Fast API, Colab, VS code.

- Implemented the *Stable Diffusion* model to implement a cutting-edge text-to-image generating project, making use of the Stable Diffusion pipeline to achieve the best possible results.
- Used fine-tuning method on Stable Diffusion model to produce high-quality images customized for specific objects.
- Using the *Dreambooth* technique, which trains on a small number of images of a subject or style, improved the text-to-image generation process' overall efficiency and creativity.

Face Swap

Tech Stack: Python, Dlib, PyTorch, face-recognition, TensorFlow, Fast API, Colab, VS code

- To ensure accurate and efficient face swapping, a model was implemented beforehand to ensure that the pose of the face in the source and target images were the same.
- Used a package named Dlib to precisely identify facial points in both images, ensuring that the features were perfectly aligned.
- Blended the faces together in a way to make sure the skin colors looked natural.

Achievements

- Ranked 2nd with the 'Weed Detection System' in BS Computer Science Final Year Project.
- Won 2nd place at the 2023 Hackathon with the 'Weed Detection System'.