Shahbaz Ali

0321-4856934 | shahbazlization@gmail.com | LinkedIn | GitHub | Updated Resume

Education

Lahore University of Management Sciences, Lahore — Lahore, Pakistan
MS Computer Science, CGPA 3.17 — *July 2019 - May 2021*
Courses Taken: Machine Learning, Design and Analysis of Algorithms, Deep Learning, Computer Vision, Advanced Operating System, Applied Probability, Digital Image Processing, Computer Architecture, Speech Processing, ICT4D

Government College University, Lahore — Lahore, Pakistan
BS Computer Science, GPA 3.43 — *Aug. 2014 – May 2018*
Courses: Operating Systems, Data Structures, Analysis Of Algorithms, Artificial Intelligence, Machine Learning, Networking, Databases, Theory of Automata, Compilers Construction

Experience

Technical Content Engineer — Feb 2021 - July 2021
Educative, Inc — *Lahore, Pakistan*
- Created/Managed difference courses on the platform
- Played part in the review process of different courses
- Conducted interviews and helped in the hiring process

Teaching Assistant — Jan 2020 - June 2021
Lahore University of Management Sciences — *Lahore, Pakistan*

Designed and graded assignments/quizzes for the following courses:
- CS-623 Hardware Architecture for AI with Dr. Rehan Hameed (SPRING 2021)
- CS-5317 Deep Learning with Dr. Murtaza Taj (SPRING 2021)
- CS-510 Design & Analysis of Algorithm with Dr. Imdad Ullah Khan (FALL 2020)
- CS-535 Machine Learning with Dr. Agha Ali Raza (SPRING 2020)
- CS-331 Artificial Intelligence with Dr. Mian Muhammad Awais (SPRING 2020)

Software Development Engineer — July 2018 - November 2019
Skills Knight Studios — *Lahore, Pakistan*
- Worked on different cross platform mobile games. Mostly added features or updates to the games already published on Play Store and App Store.
- Maintained a match 3 game with large user base. Removal of bugs reported by users and also quarterly major/minor feature updates

Projects

Inference for CNN model in C | *C/C++* — March, 2021
- Trained the CNN on fashion-mnist dataset in Keras
- Saved the weights in binary files
- Then used this weights to make prediction in C code
- Implemented convolution, fully-connected, dropout, maxpool, relu and softmax layers in C

Face vs No-Face Image Classification via Linear Classifier | *Python, Keras* — Jan, 2021
- Gather face images (male, female, child) from different datasets
- Gathered no-face images from IMAGENT data set
- Simply trained a liner classifier (without non-linearity)

Male vs Female Image Classification via CNN | *Python* — Feb, 2021
- Gathered small male and image dataset from internet
- Trained different NN and CNN architecture and compared the results
- Used different training and test data, then improved the accuracy using data augmentation

Deep Convolutional Generative Adversarial Network (DCGAN) | *Python, Keras* April 2020

- Assignment of Deep Learning course implemented using Convolutional Layers
- Generated images of emojis (with good results) from Apple Emojis Dataset
- Designed both generator and discriminator networks

kNN Classifier | *Python* July 2020

- kNN classifier implemented on Iris Data Set

Naive Bayes Classifier | *Python* June 2020

- Multi-class classification using Naive Bayes on "Twitter US Airline Sentiment" dataset.

Recurrent Neural Network (RNN) | *Python, Keras* May 2020

- Cleaning the dataset by removing stop words, punctuation, and html tags
- Positive/Negative classification of movie reviews from IMDB dataset

Frequency Domain Filtering | *MATLAB* Nov 2020

- Created loop-based and vectorized implementation for FFT and IFFT
- Applied idead, butterworth and gaussian filter in frequency domain, and compared results

Content Based Image Retrieval (CBIR) | *MATLAB* Oct 2020

- Created the database of training images histograms
- Compared and retrieved the images based on similar histograms
- Also tried equalized image histograms to see if that is a good feature to compare images. Results were negative.

File System (Linux) | *C* May 2020

- Basic file-system which has all basic functions like open(), close(), read(), write(), format(), unlink()
- Managed multiple users accessing the same files simultaneously

Memory Management (Linux) | *C* April 2020

- Mimic the functionality of malloc() and free() in C without using any external API
- Added functionality like expand, coalesce and release for more efficient Memory Management

Web Server (Linux) | *C* March 2020

- Primitive Multi-threaded Clients and Multi-threaded Server model
- Clients send a request(using socket programming), which is completed and acknowledged by Server

Hidden Markov Model (Bakis Model) | *Python* Nov 2019

- Applied HHM for part-of-speech prediction in natural language

Binary Independence Model (BIM) | *Python* Oct 2019

- Applied BIM on corpus of 25000 news articles
- With trained BIM, the retrievals were very fast (milliseconds)

Addictive Gem Match Mania (Mobile Game) | *C#, Unity Engine* Dec 2018 - Nov 2019

- Match 3 mobile game for Android and iOS
- Added core features to the game
- Also fixed user reported bugs in the game (with 100000+ users)

Clothing Classification (Neural Network) | *MATLAB* June 2018

- Implemented & designed the architecture of Neural Network from scratch
- Implemented Backpropagation algorithm to learn to learn the best color combination of clothes
- Semester Project for Computer Vision Course

Handwritten Digit Classification (Neural Network) | *MATLAB* March 2018

- Implemented & designed the architecture of Neural Network from scratch to work on MNIST Dataset
- Implemented Backpropagation algorithm to learn to learn the best color combination of clothes
- Semester Project for Machine Learning Course

Among the Dead Ones! (Desktop Game) | *C#, Unity Engine* Feb 2018 - July 2018

- FPS Survival Shooter game for Desktop platform, worked as a part of my Final Year Project for Undergraduate
- Designed and programmed AI Zombies (NPC) for the game.

- Extensive work on Unity NavMesh System in collaboration with Unity's Macanim System to control the root motion of NPC Zombies
- Slide-free and accurate pathfinding for zombie character
- Ragdoll System to detect bullets with re-animation feature for Zombies & Audio Collection System with Scriptable Object

CrickSick Scoring Application (Android Application) | *Android Studio* May 2017

- Application for scoring/recording cricket matches
- Simple User Friendly interface to detect all the event of cricket match
- Semester project for Software Engineering Course

Technical Skills

A very good understanding of OPP concepts
Familiarity and understanding of relational databases like SQL
Good in logical and structural thinking
Languages: Python, C, C++, MATLAB, C#, Java
Frameworks: Keras, TensorFlow, PyTorch
Developer Tools: PyCharm, Spyder, Jupyter Notebook, Git, VS Code, Visual Studio, Unity, Android Studio
Libraries: pandas, NumPy, Matplotlib, OpenCV, Scrappy