SCALEDGE

Diya Patel

PROFESSIONAL SUMMARY

- **~3 years** of experience in Python **Software Development.**
- Experience in **C, C++,** and **Python** language.
- Working experience in frameworks like **CNN, RNN, PyTorch**, **Darknet, U-Net and Yolov3, SSD, TensorFlow object detection API.**
- Python libraries **PyAudio, NLTK, TensorFlow, NumPy, Pandas, OpenCV, Librosa, Keras, Torch, scikit-learn.**
- Experience in processing **CSV, XML** and **XLS datasheets** using **Python.**
- Experience in Python modules **OS, Sys, Math, Time, Json, Lxml, Requests, Threading.**
- Good knowledge in Machine learning tools like **Google colab Notebook, Spyder, Jupyter Notebook, Orange Canvas, MATLAB.**
- Experience in **Linux Operating System** in complex environments.
- Experience in **creating, implementing and maintaining** scripts for process **automation**.
- Working Experience with **Jenkins, GIT** and **JIRA.**
- Working experience with **GitLab** and **Gerrit**.
- **Groovy Scripting** to develop **Jenkins pipelines.**
- **Pipeline Jenkins job** configurations and usage.
- Good knowledge of **CI/CD automation**.
- Experience in **unit testing** with ML models.
- **Python scripts** to launch any **tool commands (Shell)**.
- Good Experience with **PEP-8** and **Google coding** Guidelines in python.
- Basic knowledge in **MySQL, SQLITE, Firebase, MongoDB** database.
- Experience in **Flask Web Framework.**
- Working Experience in embedded boards such as **Raspberry Pi, Nvidia Jetson Nano.**

PROFESSIONAL EXPERIENCE

Company: Scaledge | **Embedded Engineer** | **Jun 2019 – Till date**

PROJECT DETAIL

Project: Object and Voice Recognition

Description: The main objective focuses on object and voice recognition to detect object instances of semantic objects of a certain class (such as lane, potholes, traffic sign, humans) in digital images and videos so that it can be useful in real time to be deployed on edge devices. To make a prototype which control according to the real-time conditions using analyzing the voice for recognizing various instances of class (such as sirens used on emergency service vehicles and horn) and image processing technique using OpenCV module in python and further deep learning to achieve faster processing in less time with low end devices implementation techniques like CNN, YOLO and Single Shot Detection (SSD).

Roles & Responsibilities:

- Involved in **Convolution neural networks** for **image classification.**
- Involved in **Librosa** library in python for **voice analysis.**
- Firstly, we gathered data of voice**,** train **model and applied** classification using **Convolutional neural networks**.
- **Involved** in preparing a dataset to train the model, for that **annotated** all images.
- Prepared a **python script** to increase existing dataset by **image augmentation** (flipping, color enhancement, PCA, zooming, shearing)
- Developing a low-level interface using **Python**.
- Supported team members for technical queries.
- Integration of object and voice recognition instances using **Deep Learning Framework**.
- Maintain the code using **GitLab.**
- **Debugging** and **bug fixing.**

- Developer **level testing.**

Working Tools & Technologies: **Python (Librosa, NLTK, TensorFlow, Keras, NumPy, DarkFlow), GitLab, Jenkins**
Platform & OS: **Ubuntu, Jetson nano**

Project: **Student Alertness System**

Description: The main objective is to focus on the student to check the alertness in the online classes. The framework used for detecting the face, face emotion, movement of the body, eyes blinking detection, yawn detection, and detecting the spoof person using CNN, TensorFlow, PyTorch and OpenCV.

Roles & Responsibilities:

- Listing down **features**, **scenarios.**
- Created **Flowchart** using **draw.io.**
- Collected a **dataset** and created our own dataset.
- Created **Annotating images** of dataset.
- Implemented **training script** using **python**.
- Created custom **architecture** using **CNN, TensorFlow** and **PyTorch** for training.
- Trained model using Google **Colab** notebook (GPU).
- Detection of eyes and lips, another **landmark of the face** using **OpenCV Algorithm**.
- Implemented using **Linux** environment to **automate python script** for **detection**.
- **Tested** the model.
- Creation of **Jenkins** File and pipelines.
- **Debugging** and **bug fixing.**
- Developer **level testing.**
- **Maintain** the code.

Working Tools & Technologies: **Python (TensorFlow, Keras, OpenCV), Colab, Jupiter, draw.io, GitLab, Jenkins, JIRA**
Deep Learning Framework: **TensorFlow, PyTorch**
OS: **Linux**

Project: **Advanced Driving Test System**

Description: The goal is to provide an end-to-end solution for autonomous driving tests using video analysis and machine learning techniques. The driving test system consists of different kinds of tests. The system gives the result of a driving test using detection of vehicle and lanes line (Fixed Trajectory of test area) captured from a fixed position camera. This also includes several constraints like identifying for foul i.e., vehicle should not touch the lane and must complete within predefined time limit. The project is developed using Python and a deep learning framework like You only look once (YOLO) and Single Shot detection (SSD). Edge device is also used in this project.

Roles & Responsibilities:

- Implemented **automation python script** to **manipulate** with **huge number** of **images** in directory.
- Implemented **automation python script** to modify context of **annotated xml** files.
- Developed **automation python script** that captures a **snap** from **live stream** and **extracts** the **start line** and **lane line** and **sends** it in **thread** to the **GUI server**.
- Developed **script** which runs the all-automation **python script** on **boot up**.
- Involved in **collecting datasets** for **1 class that** collected videos from **positioned cameras** then extracted **the frames** from it.
- Modified **configuration files** of **tiny YOLO** and **Mobile-SSD v2** to train the model.
- Implementation using **automation python script** for detection of vehicle from top view with angle.
- Check for different inference in edge devices and use the most suitable framework for vehicle detection.
- **TensorFlow model** conversion which is convenient in **edge devices** in order to **increase** the **computation speed.**

Working Tools & Technologies: **Python, NumPy, TensorFlow, OpenCV**
Platform & OS: **Ubuntu, Jetson nano**

Project: **Radiology images analysis using Deep Learning**

Description: The objective is to focus on the Radiology images like X-Ray, PET, and MRI images to analyze and detect lung cancer, Bone Fracture as well as tumor detection. Using deep learning frameworks like U-net, TensorFlow, and yolov3.

Roles & Responsibilities:

- Collected **dataset**.
- Automated **python** script to manipulate with **huge number** of images in the directory.
- Created **Annotating images** from the dataset.
- Implemented **automation python script** to modify context of **annotated** xml files.
- Train the Model using **Linux virtual python** environment.
- Modified **configuration files** of **Yolov3** and **U-net** to train the model.
- Implementation using **automation python script** for **detection** of lung cancer, Bone Fracture and Tumor Detection.
- **Tested** the model.
- **Integrating** Jenkins files.
- **Debugging** and **bug fixing.**
- Developer **level testing.**
- **Maintain** the code.

Working Tool & Technologies: **Python (TensorFlow, Keras), Jenkins, GitLab, JIRA**
Deep Learning Framework: **U-Net, Yolov3**
OS: **Linux**

Project: **Remove Bell Sound Noise in Voice Data**

Description: The aim is to remove the bell sound in the voice data using Audio processing and deep learning techniques. The application was created using Android Studio and deployed on it using python script.

Roles & Responsibilities:

- Implemented **automation python script** to create spectrogram.
- **U-Net** and **TensorFlow** for **spectrogram images.**
- **Librosa** library in python for **voice analysis.**
- Collected data of a **mix of voice** and **clean voice data**.
- Modified **configuration files** of **U-Net** to train the model.
- **Trained model** and applied **detection** using **Convolutional neural networks**.
- Implementation using **automation python script** for **removing bell sound noise**.
- Creating **connection** between **android application** and **python script**.
- Implementation of **android application** and **deploy** the model.
- Creating and managing **Jenkins** Pipelines.
- **Debugging** and **bug fixing.**
- Developer **level testing.**
- **Maintain** the code.

Working Tools & Technologies: **Python (Librosa, TensorFlow, Keras), C++, Kotlin, Jenkins, GitLab, JIRA**
Deep Learning Framework: **U-Net**
OS: **Linux, Android**

Project: Gunshot Detection

Description: The main purpose is to detect gunshot sounds in urban areas. To analyze the real-time environmental gunshot sound, we used Audio processing techniques like Librosa library in python and some deep learning techniques.

Roles & Responsibilities:

- Implemented **automation python script** to create **spectrograms**.
- **Convolution neural network** for spectrogram images.
- **Librosa** library in python for **Environmental Sound analysis**.
- Collecting a **data set** of Environmental sound.
- Implemented **automation python script** to **train** the model using **Keras** libraries.
- Implementation using **automation python script** for detection of gunshot voice.
- **TensorFlow model** conversion which is convenient in **edge devices** in order to **increase** the **computation speed.**
- **Debugging** and **bug fixing.**
- Developer **level testing.**
- **Maintain** the code.

Working Tools & Technologies: Python (Librosa, TensorFlow, Keras), GitLab, Jenkins, JIRA
Platform: Raspberry Pi, Jetson nano
OS: Linux

Project: Resume Ranking System

Description: The main objective is to focus on Ranking the resume. Scanning all resumes and collecting the data and stored in CSV. Preprocesses data using pandas and NumPy. The word2vec algorithm is used to Rank the resume.

Roles & Responsibilities:

- Scanned **Resumes** and collecting needed information.
- Created **Flowchart** using **draw.io.**
- Implemented **python script** for **flask connection** to webpage.
- Creating **Webpages** using **HTML and CSS.**
- Processing and Maintaining **CSV** files using **Pandas** and **NumPy**.
- Implemented **automation python script** to **train** the model using **RNN**.
- Design **Ranking Algorithm** using **Machine Learning.**
- **Debugging** and **bug fixing.**
- **Maintain** the code.

Working Tools & Technologies: Python (NumPy, pandas, NLTK, word2vec), NLP, GitLab, Jenkins
Web Framework: Flask

Project: Document Conversion

Client: Lynx

Description: The main goal of DocBook is to convert different formats of files that can be automatically derived. This project intends to convert different formats of files like FrameMaker, PDF, and Word files to DocBook. Also, usage of various style sheets to convert DocBook to PDF.

Roles & Responsibilities:

- Used **FrameMaker** tools to convert FrameMaker files to xml.
- Used **open-source library Pandoc** to convert xml files to DocBook files.
- **Manually** verified all formats of the files.
- Created **environment** for converting DocBook files to PDF.
- Created **automated scripts** to run and build the environment for the system.
- **Testing** the target files.

- Create, collaborate and organize all our work in **confluence.**
- **Maintain** the source file code using **Gerrit.**

Working Tools & Technologies: **Python, FrameMaker Tool, Gerrit, Confluence**
OS: **Linux and windows**

TECHNICAL SKILLS

Programming Languages:	**Python (Core, OS, Sys, Math, Time, Json, Lxml, Requests, Threading, Scikit-Learn, Librosa, PyAudio, NLTK, NumPy, TensorFlow, OpenCV, Keras, Pandas), C, C++, JAVA(Basic), Golang, Shell script.**
Tools Used:	**Google Colab notebook, Spyder, Jupyter Notebook, Orange Canvas, draw.io, ANDROID studio, MATLAB, PyCharm, Labeling, FrameMaker**
Databases:	**MySQL, SQLite, Firebase, MongoDB**
Frameworks:	**CNN, Darknet, RNN, PyTorch, Android, U-Net, Yolov3, TensorFlow API, Flask**
Operating System:	**Windows, Linux, Android**
CI/CD Tools:	**Git, Jenkins and JIRA**

EDUCATION

- M. Tech. in Computer Engineering from **Dharmsinh Desai University**, Nadiad,2020 with 7.1/10.
- B.E in Computer Engineering from **SVBIT** College, Gandhinagar, **Gujarat Technological University**, 2018 with 8.53/10.0