Github **LinkedIn**

Nirbhay Sharma

9369630713

Email **Portfolio**

Education

B.Tech, CSE | IIT Jodhpur Aug'19-May'23

CGPA(Ongoing): **8.76**/10

Class 12^{th} | Dehradun public school Mar'18-Mar'19

Percentage: **96.4**

Technical Skills

Languages: Python, C/C++, HTML/CSS, Javascript, Haskell, Prolog

Tools and Frameworks: Pytorch, Sklearn, Numpy, Pandas, Matplotlib, Seaborn, Flask, Django, Regex, Heroku, Git, Github, Firebase, Mongodb, Mysql

Familier with: Tensorflow, Java, React, Nodejs, ejs, Google Colab, OpenCV

Experience

Split Neural Networks | Pytorch, Python, Jetson Nano, PySyft

Intern (ML Engineer) | **Exawizards** India Jun'22-July'22

- Splitted **Mask-RCNN, FCN_Resnet50, YOLOv5** for **Instance segmentation, segmentation, face detection** tasks respectively
- Used **Pysyft** and **Jetson Nano** for transfering features from one device to another
- Used **encoder-decoder** architecture for **tensor compression**
- Reduced **inference time** on Jetson nano device while preserving **data privacy**

Light Weight CNN Model for Chest Radiographs Classification | Pytorch, Python, Torchvision, Numpy

Mentor: Dr. Angshuman Paul | IIT Jodhpur Jun'21-Mar'22

- Designed a **Lightweight CNN model** for the abnormal detection of **Chest Radiographs**
- Combined the ideas from **Squeezenet** and **Mobilenet** to prepare a Light weight model
- Model Outperforms various light weight CNN architectures like **Squeezenet**, **Shufflenet**, **Mobilenet** on NIH dataset both on binary and multiclass classification

Projects

CNN Algorithms Comparison | **Github** | Pytorch, Numpy, Matplotlib, PIL, Python

- Compared **7** deep **CNN** architectures on **Retinal Eye disease dataset**
- Coded **Squeezenet**, **Mobilenet**, **Inceptionnet**, **Shufflenet**, **Googlenet**, **Resnet**, **Efficientnet** from **scratch**
- Performed a comparison study among the state-of-the-art deep CNN architectures

Image Colorization | **Github** | Pytorch, Numpy, Matplotlib, PIL, Python

- Converted **grayscale image** to **colored image** using **GAN** architectures
- Used **pix2pix** GAN from scratch for the colorization task
- Performed colorization on **LAB** and **RGB** image format

Mask-NoMask Detection | **Github** | Pytorch, Numpy, PIL, Matplotlib, Python

- Detected **5300** images under masked and no masked category with an accuracy of **99.6%**
- Used transfer learning with **Mobilenet v2** for classification task
- Combined the trained model with **OpenCV** for real time classification

PRA-Visualizer | **Github** | **Url** | React, Nodejs, HTML, CSS, Firebase

- Implemented a **Page replacement algorithm visualizer** which simulates various page replacement algorithms given **Frames** and **demand pages**
- Implemented **10** algorithms including **LRU**, **Working set**, **FIFO** etc.