Sudarshana Lakshmi Krishna

(+1) 919-396-4568 | skrish28@ncsu.edu | slakshkrish

Education

North Carolina State University, Raleigh

M.S. in Electrical Engineering *August 2019 - May 2021*

GPA: 3.667

Courses: Digital Imaging Systems, Digital Signal Processing, Random Processes, Computer vision, Neural Networks, Artificial Intelligence, deeplearning.ai(Coursera)

Easwari Engineering College, Chennai, Tamil Nadu, India

B.E. in Electronics and Communication Engineering *August 2015 - May 2019*

Skills

Programming Languages	Python, C/C++, LaTeX, MATLAB
Softwares	Arduino Software(IDE), MATLAB App Designer, VSCode, Microsoft Office, Jupyter notebook
Libraries & Toolboxes	MATLAB Image and Signal Processing ToolBox, numpy, scipy, openCV, scikit-learn, pandas
Deep Learning Platforms	Tensorflow, Keras
OS	Windows, Linux

Experience

North Carolina State University

Student Researcher - Domain: Computer Vision - Object Detection *Present*

- Working under Prof. Dr. Tianfu Wu on Auto Context Region based Convolutional Neural Networks(R-CNN).
- Working on improvising the available opensource mmdetection toolbox for auto context based object detection using state of art methods of RoIPooling and RoIAlign.

Image Processing and Computer Vision(IPCV) Lab, Indian Institute of Technology, Madras(IITM), India

Summer Intern - Domain: Image Processing - Image Segmentation *May 2018 - July 2018*

- Worked under Prof. Dr. A.N.Rajagopalan on the project **Dirt Detection on Camera Lens**.
- Devised an algorithm using MATLAB to detect regions of dirt on camera lens from captured images.
- Achieved the above through a combination of spatial gradient, spatial variance, temporal intensity differences and superpixel segmentation.
- Accomplished successful detection for solid dirt and water droplets.

Projects

Hyperparameter Tuning: Study of Babysitting Process for a Deep Learning model

Domain: Computer Vision- Classification Model Optimization

- Using the CIFAR 10 dataset on Keras, I analysed the model performance for both Multi Layer Perceptron(MLP) and Convolutional Neural Network(CNN) model.
- Studied changes in model behviour by varying different hyperparameters like optimizer, decay rate, momentum, learning rate scheduler, filter size, dropout values.
- Achieved a final testing accuracy of 88% on CNN model.

Spam and Ham classification

Domain: Natural Language Processing (NLP)

- Used the 'SMS Spam Collection Data Set' from UC Irvine's Machine Learning Repository.
- Implemented a Naive Baye's Classifier for classification between Ham and Spam messages.
- Coded the model on Google Colab using Python 3.6 by calling functions for the Naive Baye's classifier from scikit-learn.
- Achieved a precision of 94%.

Classification of Mosquito Species based on their Wing structure

Domain: Computer Vision- Image Classification

- A team project aiming to distinguish between restuans and pipiens species of the Culex mosquito variety.
- Implemented a transfer learning model of VGG19 trained on ImageNet to classify the species based on images of their wings.
- transfer learning performed poorly on the data set giving a validation accuracy of 49% and testing accuracy of only 25%.
- Built the final CNN model mimicking a few layers from VGG. This gave an accuracy of 68%.

Identification of Leaf Wilting stages in Plants

Domain: Computer Vision- Image Classification

- Pre-processed data set containing crop images with five leaf wilting stages for a Convolutional Neural Network(CNN) model.
- In a team of 3 members, I prepared the data set applying data augmentation techniques using openCV in Python 3.6.
- Performed flip, crop, gaussian blur, image pixel enhancement methods to increase the data set.
- The model gave an overall accuracy of 47%.

Face classification using AdaBoost

Domain: Computer Vision - Machine Learning - Image Classification

- Built the AdaBoost algorithm from scratch using Python 3.6 on VSCode platform using haar-ike features for weak classifiers.
- Using the type-2-y and type-3-x haar features 10 best features were compiled during training.
- The training images were of dimensions: 16x16x3, while it was tested on images of 20x20x3 dimensions.
- Evaluation using ROC plot places the model better than random guess.