Sudarshana Lakshmi Krishna

919-396-4568 | skrish28@ncsu.edu | slakshkrish

Education

North Carolina State University, Raleigh *August 2019 - May 2021*

Master of Science in Electrical Engineering, GPA: 3.667

Courses: Digital Imaging Systems, Digital Signal Processing, Random Processes, Computer vision, Neural Networks, Artificial Intelligence, deeplearning.ai(Coursera), Data Science(Coursera- from John Hopkins University), Machine Learning(Coursera- from Stanford), NLP(Coursera)

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 *Present*

Student Researcher - Domain: Computer Vision - Object Detection

- Took lead individually on project 'Auto Context Region based Convolutional Neural Networks(R-CNN)' under Prof. Dr. Tianfu Wu.
- Incorporating open source 'mmdetection toolbox' features to re-create auto context based object detection.

Image Processing and Computer Vision(IPCV) Lab, Indian Institute of Technology, Madras(IITM), India *May 2018 - July 2018*

Summer Intern - Domain: Image Processing - Image Segmentation

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

Projects

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

Domain: Computer Vision- Classification Model Optimization

- Analysed model performance for both Multi Layer Perceptron(MLP) and Convolutional Neural Network(CNN) model by testing with CIFAR10 dataset on Keras.
- Studied changes in model behaviour by varying different hyperparameters - optimizer, decay rate, momentum, learning rate scheduler, filter size, dropout values.
- Procured a final testing accuracy of 88% on CNN model.

Spam and Ham classification

Domain: Natural Language Processing (NLP)

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

Classification of Mosquito Species based on Wing structure

Domain: Computer Vision- Image Classification

- Contributed in a team project aiming to distinguish between Restuans and Pipiens species of Culex mosquito variety.
- Implemented a transfer learning model of VGG19 trained on ImageNet to classify mosquito species based on wing structure as trial.
- Above procedure performed poorly on data set giving a validation accuracy of 49% and testing accuracy of only 25%.
- Built a final CNN model mimicking a few layers from VGG leading to an accuracy of 68%.

Identification of Leaf Wilting stages in Plants

Domain: Computer Vision- Image Classification

- Project involves classification of plant images into 5 different stages based on leaf wilting levels.
- Shared ideas in an inter-cultural team of 3 members from different countries.
- Pre-processed data set containing crop images with five leaf wilting stages for a Convolutional Neural Network(CNN) model.
- Took initiative in preparing data set and applying data augmentation techniques using openCV in Python 3.6.
- Performed flip, crop, gaussian blur, image pixel enhancement methods to increase data set variability and size.
- Model created,gave an overall accuracy of 47%.

AdaBoost based Face Classification

Domain: Computer Vision - Machine Learning - Image Classification

- Built AdaBoost algorithm from scratch using Python 3.6 on VSCode platform, exercising haar-ike features for weak classifiers.
- Compiled 10 best type-2-y and type-3-x haar features during training.
- Training images were of dimensions: 16x16x3, while it was tested on images of 20x20x3 dimensions.
- Evaluated model proved to be higher than random guess when tested with ROC plot as metric.