

Vishal Asnani



V-card

Phone: +1 (517) 505-4249

Email: asnani@msu.edu

LinkedIn: www.linkedin.com/in/vishal-asnani

GitHub: <https://github.com/vishal3477>

OBJECTIVE

To put my skills to use in exciting, innovative ways while working for something greater than myself and gain some in-depth ideas in the field of Computer Vision and Deep Learning.

EDUCATION

Ph.D. In Computer Science and Engineering
Advisor: Dr. Xiaoming Liu
Michigan State University, East Lansing, USA
GPA: 3.8/4.0

| Jan. 2021-Current|

M.S In Electrical and Computer Engineering
Michigan State University, East Lansing, USA
GPA: 4.0/4.0

| Aug. 2019-Dec. 2020|

B.E.(Hons.) Electronics and Instrumentation Engineering (Minor in Finance)
Birla Institute of Technology and Science, Pilani, India
CGPA: 8.01/10.0

| Aug. 2015- May 2019|

PUBLICATIONS

- **Vishal Asnani**, Xi Yin, Tal Hassner, Sijia Liu, Xiaoming Liu, "Proactive Image Manipulation Detection," In Proceeding of IEEE Computer Vision and Pattern Recognition 2022.
- **Vishal Asnani**, Xi Yin, Tal Hassner, Xiaoming Liu, "Reverse Engineering of Generative Models: Inferring Model Hyperparameters from Generated Images" under review in IEEE Transactions on Pattern Analysis and Machine Intelligence 2022.

TECHNICAL PROFICIENCY

Tools, Simulation, and Software Platforms

- TensorFlow, PyTorch, Keras, Numpy, Scikit-learn, Jupyter, OpenCV, CUDA, MATLAB and Simulink, GCS, Amazon Web Services (AWS), LabVIEW, Linux, CST Microwave studio, Cadence virtuoso, Microsoft-Visual Studio, Excel, Word and Powerpoint, Orcad PSpice, Labcentre Proteus, Eagle- PCB Design and Schematic Software, Xilinx Vivado Suite and SDK, FluidSim

Languages / Scripts

- Python, MATLAB, R, SQL, Perl, C, C++, Cascading Style Sheets (CSS), JavaScript, HTML, Verilog, VHDL, x86 Assembly Language, Arduino Programming

PROJECTS

1. **Proactive scheme for image manipulation detection by adding learnable templates** |Jan. 2021-Nov. 2021|
 - A **novel proactive scheme** is proposed which encrypts a real image by adding a template from a **learnable template set**.
 - The **added template** is later **recovered** to perform image manipulation detection.
 - The template set is **learned** using defined **constraints** which incorporate properties including small magnitude, more high-frequency content, orthogonality, and classification ability
 - **Near-perfect** average precision is obtained for unseen Generative Models (GMs) compared to prior works.
 - The proposed framework is **more generalizable** to different GMs, showing an **improvement of 10%** average precision averaged across 12 GMs compared to prior works.
2. **Model Parsing: Reverse engineering of hyperparameters of generative models** |Jul. 2020-Apr. 2022|
 - A **novel** problem of **Model Parsing** is defined to develop a framework for **predicting** the **network architecture** and **loss functions** given a generated image.
 - We estimate the **mean and deviation** for each GM using **two different parsers**: cluster parser and instance parser which are then combined as the final predictions.
 - A network architecture super-set with **15 features** and a loss function type super-set with **10 features** were selected to represent every GM.
 - 1000 images each for 116 generative models were collected to create a **new dataset**, and the experiments were conducted in the **leave-out** setting.
 - The framework has an **L1 error of 0.149** with a **p-value of 0.00045** for **continuous** type parameters and **71.8% F1-score** for **discrete** type parameters in **network architecture** prediction.
 - **81.3% F1-score** for the type of **loss function** prediction was achieved.

- The method generalizes well to tasks of **deepfake detection** on the Celeb-DF benchmark and **image attribution** with an AUC of **74.60%** and **99.32%**, respectively, in both cases reporting results comparable with existing **SOTA**.

- Deepfake video detection model built using PyTorch** |Feb. 2020-Apr. 2020|
 - Implemented a machine learning model consisting of Convolution neural networks (CNN) followed by a recurrent neural network (RNN) for **deepfake detection**.
 - The **CNN-RNN model** would be able to detect whether a video is fake or real.
 - The frames were extracted using the MTCNN model, which was then passed into the CNN-RNN model, **trained** on the **Face-Forensics++** (FF++) dataset, and **tested** on the **FF++ and Celeb-DF** datasets.
 - The model achieves **98.2% AUC** on the FF++ dataset and **68.1% AUC** on the Celeb-DF dataset.
- Machine learning classifiers to classify news into different categories.** |Mar. 2020-May 2020|
 - **News category dataset** available on Kaggle was used to test **different classifiers classifying news** into different categories.
 - **Many classifiers** were tested like Multinomial Naïve Bayes, Multinomial Gaussian Bayes, K nearest neighbor, Decision trees, Linear Support vector machines, Multi-class logistic regression, Bagging, AdaBoost, and Recurrent neural network.
 - The models were tested on the **different trains: test split ratio** and **different hyperparameters** for each classifier.
 - Multinomial Gaussian Bayes, Multi-class logistic regression, and Linear Support vector machines performed better than other classifiers with an **accuracy of 75%**.
- Framework for information retrieval using PyTorch** |Aug. 2019-Dec. 2019|
 - Developed a framework for **binding and retrieving class-specific information** from image patterns using correlation filters.
 - The template-based framework works by **matching the template** with the **query pattern** of the same class.
 - Class-specific information is mapped to spatial translations applied to image patterns to generate a multi-peak correlation filter.
 - The **CMU PIE and FRGC databases** were used to study the relationship between the False class Information retrieval rate (FCIRR) and several peaks of the correlation filter.
 - FCIRR had an **inverse relationship** with several peaks. The **FCIRR for 30 peaks** was **0.02 for CMU PIE and 0.88 for FRGC**.
- Deep Learning model involving modified version of LSTM RNN** |Jan. 2019-May 2019|
 - Developed a deep learning framework involving two parameter-reduced variants of the **LSTM layer**.
 - The model was built using high-level Keras API in the Jupyter Notebook with the **UCI News Aggregator dataset**.
 - **1% increase** in accuracy was successfully achieved in the model involving a modified LSTM layer compared to the standard LSTM layer.

WORK EXPERIENCE

Texas Instruments, Bengaluru, India

| Jul. 2018- Dec. 2018 |

- Analog design intern in **Multiphase and Control solutions team**.
 - Developed a Perl Script to create vector-based patterns for SWD and PMBus commands used in the test program.
 - Patterns were appended to the test program beforehand, thereby saving test time.
 - Efficient implementation of **SWD and PMBus patterns** saved approximately **74% execution time**.
-

RELEVANT COURSES PURSUED

- Computer Vision: Detectors and Descriptors, Optical Flow, Image segmentation, Tracking and object detection in videos, Epipolar geometry.
 - Machine learning: Regression, Classification, Dimensionality reduction, Sparse learning, Matrix competition, Ensemble methods, Multi-task learning
 - Pattern recognition and analysis: Bayesian classification, Estimating gaussian MLE parameters, Non-parametric density estimation.
 - Deep Learning: Deep Neural Networks, Convolution Neural Networks, Recurrent Neural Networks, Sequence Models.
 - Deep Learning specialization- deeplearning.ai by Andrew Ng (Coursera).
-