SAIKAT BASAK

A-8/303, Kalyani, WB, India, Pin. 741235

(+91) 9163972637

 saikatbsk@gmail.com

 in.linkedin.com/in/saikatbsk

 github.com/saikatbsk

ABOUT ME

"What I cannot create, I do not understand." – Richard Feynman

Driven by the passion for science and technology. Keen in tinkering, breaking and creating. Resourceful, willing to make mistakes and learn. Believer of the Church of GNU and Software Freedom. Avid Sci-Fi fan and gamer by heart.

I describe myself as a skeptic. I believe in reason and critical thinking.

The prospect of Artificial Intelligence fascinates me. I wish to see general artificial intelligence become a reality and hope to be one of the architects of the future of AI. I also have an OCD for symmetry.

EDUCATION

Master in Multimedia Development (M.E. Equivalent) — Expected June 2017
Jadavpur University

Master of Computer Applications — 2011 – 2014
West Bengal University of Technology

Bachelor of Science (Physics Honors) — 2008 – 2011
Vidyasagar University

(Continued..)

PUBLICATIONS

Journal Publications

Saikat Basak, Ranjan Parekh, "**An Improved Bag-of-Features Approach for Object Recognition from Natural Images**", International Journal of Computer Applications (IJCA), 2016.

Conference Publications

Susovan Jana, Saikat Basak, Ranjan Parekh, "**Automatic Fruit Recognition from Natural Images using Color and Texture Features**", Proc. of the 2nd international conference on Devices for Integrated Circuit (DevIC), IEEE, 2017.

Saikat Basak, Arundhuti Chowdhury, "**Gesture: A New Communicator**", Proc. of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), Springer International Publishing, 2014.

PROJECTS

Image Completion using Deep Convolutional Generative Adversarial Network

Adversarial training is a way to train two neural networks simultaneously. The first one is the Discriminator that takes an input (e.g. an image) and outputs a scalar indicating whether the image looks "natural" or not. The second network is a Generator. The role of this generator is to generate a fake image so as to train the Discriminator to output the correct probability of it being "fake" or not. During the process, the Generator network learns to generate natural looking images (face images in this case) and the Discriminator learns to distinguish between the natural and fake images. I'm applying this DCGAN architecture in image completion/inpainting.

Implemented in **Python** using **TensorFlow**.

URL: https://github.com/saikatbsk/ImageCompletion-DCGAN

TensorFlow-GAN: Generative Adversarial Network

This is a basic GAN implementation with an aim to understand the core concepts behind "generative models via an adversarial process".

Implemented in **Python** using **TensorFlow**.

URL: https://github.com/saikatbsk/TensorFlow-GAN

DeepView

DeepView is an approach for classifying images using Convolutional Neural Networks. Here, I have explored model scalability, optimization techniques, activation functions and other important aspects of deep neural network architectures. The final model has currently been tested on the CIFAR-10 dataset with promising results. I plan to explore data augmentation and see how that affects the classification accuracy.

Implemented in **Python** using **Keras**.

URL: https://github.com/saikatbsk/DeepView

MNIST-CNN

CNN model trained to classify handwritten digits from the MNIST dataset. MNIST is a dataset of 60,000 28x28 grayscale images of the 10 digits, along with a test set of 10,000 images. The model achieves 99.23% accuracy which is at par with some of the current benchmarks.

Implemented in **Python** using **Keras**. Data visualization using **t-SNE**.

URL: https://github.com/saikatbsk/MNIST-CNN

Understanding-CNN

Visualizing ConvNet layers provide a deep Understanding of how neural networks see the world. Maximizing the responses from conv filters give us a neat visualization of the convnet's modular-hierarchical decomposition of its visual space. The first layers basically just encode direction and color. These direction and color filters then get combined into basic grid and spot textures. These textures gradually get combined into increasingly complex patterns. The filters become more intricate as they start incorporating information from an increasingly larger spatial extent.

Implemented in **Python** using **Keras**, **Scipy**.

URL: https://github.com/saikatbsk/Understanding-CNN

Vincent: AI Artist

Style transfer is the technique of recomposing images in the style of other images. Vincent is an implementation of the popular neural-style paper.

Implemented in **Python** using **Keras**, **Scipy**, and **Sklearn**.

URL: https://github.com/saikatbsk/Vincent-AI-Artist

(Continued..)

DeepDream

Baking my own Deep Dream. This is an implementation of the popular conv filter visualization technique known as DeepDream/Inceptionism introduced in Google Research Blog.

Implemented in **Python** using **Keras**, **Scipy**.

URL: https://github.com/saikatbsk/DeepDream

bagOfFeatures

Image classification using the Bag-of-Features model. Training images are represented using clustered and quantized Speeded-Up Robust Features (SURF). Classification using K-Nearest Neighbor. Accuracy on a subset of Caltech101 (with 10 classes) is 76.67%.

Implemented in **Octave**.

URL: https://github.com/saikatbsk/bagOfFeatures

Wave: A Perceptive User Interface

A perceptive user interface based on real-time hand detection and tracking. Requires a webcam. Useful for applications such as gesture-based point and click interfaces.

Implemented in **C++** using **OpenCV**.

URL: https://github.com/saikatbsk/Wave

ISHARA: Mouse Control with Gesture

An interface for controlling mouse pointer using finger gestures. Requires a webcam. Includes gestures for emulating click (left and right) and scroll using color markers.

Implemented in **C++** using **OpenCV**, and **Qt**.

URL: https://github.com/saikatbsk/Ishara