## 

**Colorization of B/W images using DNNs**

**Mitul Tandon**

January 2024

## 

**Week 1**: Introduction to Python and its Libraries

The incorporation of Numpy, Pandas, PyTorch, and Matplotlib played a vital role in achieving the objectives of my colorization project.

* Numpy  
   Learned Numpy for efficiently handling numerical operations, especially in array manipulation crucial for preprocessing in our colorization project.
* Pandas  
  Adopted Pandas for streamlined data handling, a robust solution for loading, cleaning, and preparing datasets, seamlessly integrating data preprocessing with subsequent project phases.
* PyTorch  
  Gaining proficiency in PyTorch proved foundational for our deep learning endeavours, enabling the creation and training of our colorization model.
* Matplotlib  
  Discovered Matplotlib and implemented it in visualizing colorization outcomes, representation of colorized images, and performance metrics.

Utilized the knowledge of the above-mentioned libraries in loading, preprocessing, and augmentation of MNIST data sets. Effectively trained a model to achieve an accuracy of 89% using only Linear layers in the Neural network.

* **Week 2**: Colorization Magic with CNNs and Advanced Techniques

This week's advancements included a PyTorch-based CNN implementation, exploration of custom loss functions, strategic transfer learning with pre-trained models, and the incorporation of batch normalization techniques.

* CNNs and its Implementation Using PyTorch  
  Thoroughly understood the concepts of CNNs, their role in the Colourization of images and their implementation using PyTorch library through several online resources.
* In-depth Study of Loss Functions

Learned about Loss functions in Machine Learning for numerous tasks. Also explored custom loss functions, focusing on perceptual losses for colorization.

* Transfer Learning

Understood the concepts of transfer learning through online resources and how it can be used in future development of the project.

* Batch Normalization Techniques

Implemented batch normalization within the CNN architecture to stabilize training, addressing issues like vanishing or exploding gradients.

Revisited the MNIST dataset, building upon the preprocessing completed in the previous week's assignment. Implemented a Convolutional Neural Network (CNN) for training, achieving an impressive accuracy surpassing 96%. Incorporated advanced techniques such as batch normalization, convolutional layers, and max pooling to optimize model performance and enhance feature extraction.

* **Week 3**: Exploring Colorization with Neural Networks
* Article Study  
  Engaged with an insightful Medium article by Emil Wallner on [Colorize B&W Photos with a 100-line Neural Network](https://emilwallner.medium.com/colorize-b-w-photos-with-a-100-line-neural-network-53d9b4449f8d). The article provided a concise yet comprehensive overview of the colorization process, serving as a foundational reference for subsequent project tasks.
* Understanding Convolutional Autoencoder Architecture  
  Acquired a thorough understanding of Convolutional Autoencoder architecture, a key concept vital for the subsequent implementation phases.
* Research Paper Exploration:  
  Delved into a research paper on the colorization of black and white images, extracting valuable insights and methodologies to inform the subsequent design of the colorization model.

**Implementation of Colorization Models**

As the culminating assignment of the project, three iterations of the colorization model were implemented using PyTorch.

**Alpha Model**Trained a machine learning model on a single image using autoencoders, achieving significant success in image retrieval. This alpha model served as a proof of concept for the proposed approach.

**Beta Model**Expanded the scope by training the model on multiple images and increasing the number of epochs. This involved the structured use of encoder and decoder components, contributing to improved colorization outcomes.

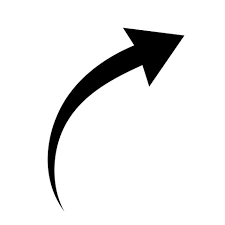
**Final Model with Transfer Learning**  
Leveraged the power of transfer learning by integrating the Inception ResNet v2, a powerful classifier trained on 1.2M images. By transferring learning from the classifier to the coloring network, the final model will gain an understanding of object representations, enhancing its ability to match objects with appropriate coloring schemes.

*The success of the alpha model, scalability in the beta model, and the integration of*

*transfer learning in the final model collectively demonstrates a progressive and*

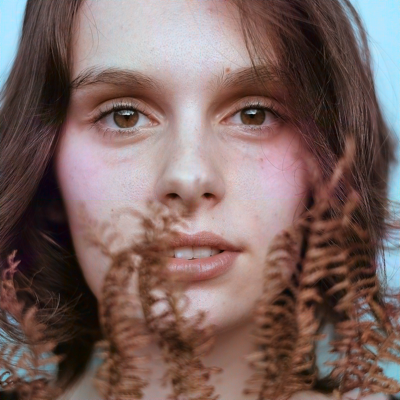
*well-rounded approach to the project objectives.*

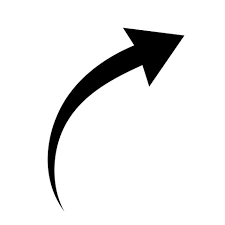
Results and Findings

** **Conversion to**

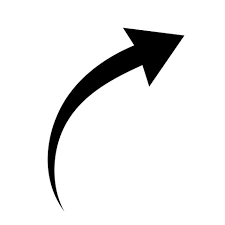
**Gray-scale image**



****

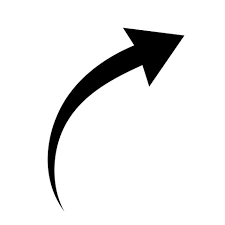


**Alpha Model**

 **Conversion to Gray-scale image**

**

**

**

**Final Model**

*As anticipated, the loss function reached saturation due to the nature of the broad training dataset. Consequently, a predominant brownish tint was observed in the majority of the images. Despite this, there were still discernible blue patches in the images, suggesting that further improvements could be achieved with an increased number of epochs and enhanced computational resources.*