

**Tvarit**  
**HACKATHON**  
**2020**

**“Become an AI-Powered Artist”**

**Neural Style Transfer**



## What is Neural Style Transfer?



Content Image

+



Style Image

Style Transfer



Stylized Output

**Problem Statement:** Develop an application to transfer **Artistic Style** from a **Style Image** to a **Content Image**

# Challenges



Content Image

+



Style Image

Style Transfer



Stylized Output

- **Challenge 1** - Develop a codebase to perform Neural Style Transfer given a content image and a style image
- **Challenge 2** - Design a web-application to have the functionality, upload own style and content images using Flask or FastAPI
- **Challenge 3** - Deploy the web-application onto a cloud platform and perform the style transfer as an on-demand function (For example - AWS Lambda function)

## Objectives

- Understand the Deep Learning Pipeline for Neural Style Transfer
- Building a framework for the style transfer given a content and a style image
  - Visualizing intuitive intermediate results for the user
- Build a web application running locally to host the application
- Deploying the application onto cloud considering resource efficiency

## Deliverables

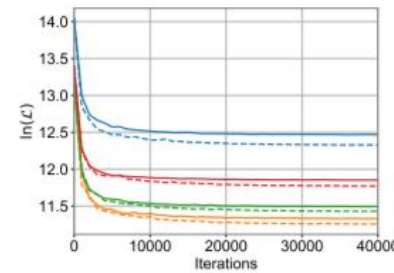
- **Deliverable 1:** GitHub Repository with a Readme
- **Deliverable 2:** A Four-minute Presentation per team at 6PM IST

## Have Fun Coding!

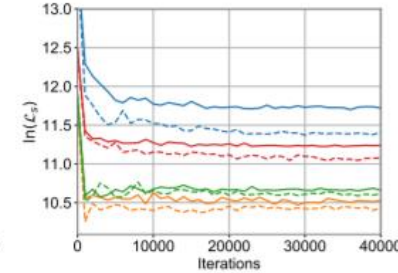
- Healthy style transfer - preserves most of the properties from the content image and takes most of the style properties
- Prepare a robust application - to work with any content/style images
- Please reach out to our *Mentors* for any queries on the Problem Statement
- Please reach out to anyone from Tvarit regarding infrastructure related questions

# Evaluation

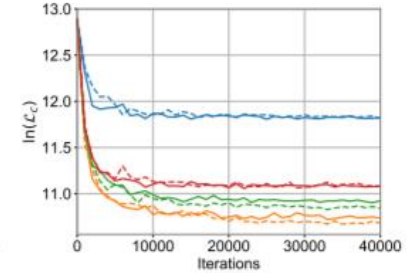
1. Presentation – What is new your approach?
2. Style Transfer - 500 iterations - Start with Content Image
  - 2.1. Loss comparison - Normalize images for training - **L2 Loss should be reported**
    - Deliverable: loss arrays as team\_name\_(i).csv
  - 2.2. L2 Distance between output image - Style image, Content Image - `sklearn.metrics.pairwise.euclidean_distances`
  - 2.3. Output Images (Visual evaluation)
3. Web application (**Bonus 1**)
  - Functionality of the UI - Upload option, Display
4. Cloud Deployment (**Bonus 2**)
  - Extra points for Lambda function



(a) Total Loss Curve



(b) Style Loss Curve



(c) Content Loss Curve