MULTI STYLE TRANSFER
A PROJECT REPORT

Submitted by
SALONI CHAUHAN (03414202017)
VIKRANT KUMAR (04014202017)
ANIKET SEHRAWAT (35214202017)

in partial fulfillment for the award of the degree

of

BACHELORS OF COMPUTER APPLICATION

JAGANNATH INTERNATIONAL MANAGEMENT SCHOOL VASANT KUNJ, NEW DELHI – 110070

May, 2020

SELF CERTIFICATE

This is to certify that the project report entitled "**Multi style transfer**" is done by us is an authentic work carried out for the partial fulfilment of the requirements for the award of the degree of BCA(2017-2020) under the guidance of **Ms. Snehlata**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of our knowledge and belief.

[SALONI CHAUHAN]	[VIKRANT KUMAR]	[ANIKET SEHRAWAT]
[03414202017]	[04014202017]	[35214202017]

Certificate

Certified that the Project Report (BCA-356) entitled "**Multi style transfer**" done by the group of students **SALONI CHAUHAN(03414202017**), **VIKRANT KUMAR(04014202017)** and **ANIKET SEHRAWAT(35214202017)**is completed under my guidance.

Date: May 30, 2020

Name of the Guide: Ms. Snehlata

Designation: Assistant Professor

ACKNOWLEDGEMENT

With candor and pleasure we take opportunity to express our sincere thanks, obligation and deep gratitude to our esteemed guide **Ms.Snehlata** *[assistant professor]* who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system**.** It is because of her able and mature guidance and co-operation without which it would not have been possible for us to complete our project.

I heartily thank for her guidance and suggestions during this project work and otherwise too.

We are also thankful to and fortunate enough to get constant encouragement, support and guidance from **Dr. Meenakshi Narula,** Head of Department of Information Technology and all other Teaching staffs of Department of Information Technology which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in computer laboratory for their timely support.

Finally, we gratefully acknowledge the support, encouragement & patience of our families, and God without their blessings and support nothing in my life would be possible.

Thank You!

[SALONI CHAUHAN] [03414202017]

[VIKRANT KUMAR] [04014202017]

[ANIKET SEHRAWAT] [35214202017]

TABLE OF CONTENTS

ABSTRACT

Whether we're aware of it or not, computer vision is everywhere in our daily lives. For one, filtered photos are ubiquitous in our social media feeds, news articles, magazines, books—everywhere!

This project explores methods for style transfer based on convolutional neural networks.

Painters have a limited number of basic pigments to mix. They paint a scene using those pigments, adding or omitting elements to make their best representation of the scene. We require to edit & improve photo. Rebalance color to make it more pleasant or more true seeming. Add or remove distortions to make things prettier or more grotesque. In this project, we use deep learning to compose one image in the style of another image. As a technique it can combine both artistic aspects and recognition (content) aspects of images

Neural style transfer uses neural networks and artificial intelligence to apply artistic effects to transform images. It is an optimization technique used to take two images, a *content* image and a *style reference* and blend them together so the output image looks like the content image, but painted in the style of the style reference image.

The algorithm takes three images, an input image, a content-image, and a style-image, and changes the input to resemble the content of the content-image and the artistic style of the style-image.

Chapter 1

INTRODUCTION

Chapter 1
INTRODUCTION

We all have used apps like Prisma and Lucid, but ever wondered how these things works? Like we give a photo from our camera roll and select a design to mix both the images and we get a new image which has the content of our input image and style of the design image.

In the world of deep learning this is called **style transfer.**

Among the applications of convolutional neural networks (CNN) and visual recognition, style transfer has been a very heated topic. As a technique that can combine both artistic aspects and recognition (content) aspects of images, style transfer has always become an interesting topic for researchers in the field of computer vision. Style transfer is essentially combining the style of one image into the content of another.

Style transfer uses the features found in the 19-layer VGG Network. It comprise of a series of convolutional and pooling layers, and a few fully-connected layers. Multi style transfer, allows you to take an image and reproduce it with a new artistic styles.

1.1 PRINCIPLE OF STYLE TRANSFER

The principle is simple: we define two distances, one for the content and one for the style. Content distance measures how different the content is between two images while style distance measures how different the style is between two images. Then, we take a third image, the input, and transform it to minimize both its content-distance with the content image and its style-distance with the style-image. Now we can import the necessary packages and begin the neural transfer.

1.2 SCOPE

- Deep learning and neural networks are undoubtedly going to fundamentally change the way we create media and art of all kinds.
- **Style transfer** is a machine learning technique for combining the artistic style of one image with the content of another image.
- The algorithm presented attempts to model this capability of humans to produce new pieces of art.
- Much progress can be made on style transfer to extend the style transfer technique from still images to videos and even audio and other mediums.
- Creating better quality art using machine learning techniques is imperative for reaching human-like capabilities, as well as opens up a new spectrum of possibilities.

1.3 OBJECTIVES

This project explores methods for artistic style transfer based on convolutional neural networks. The topic is particularly interesting because it creates artificial intelligence that inter-plays the content and the style of an image to produce artistic results of high quality. Deep neural networks have already surpassed human level performance in tasks such as object recognition and detection. However, deep networks were lagging far behind in tasks like generating artistic artefacts having high perceptual quality until recent times. Creating better quality art using machine learning techniques is imperative for reaching human-like capabilities, as well as opens up a new spectrum of possibilities.

We will follow the general steps to perform style transfer:

1. Visualize data
2. Basic Pre processing /preparing our data
3. Set up loss functions
4. Create model
5. Optimize for loss function

1.4 SDLC METHODOLOGIES

V-Shaped Model- The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model.

It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

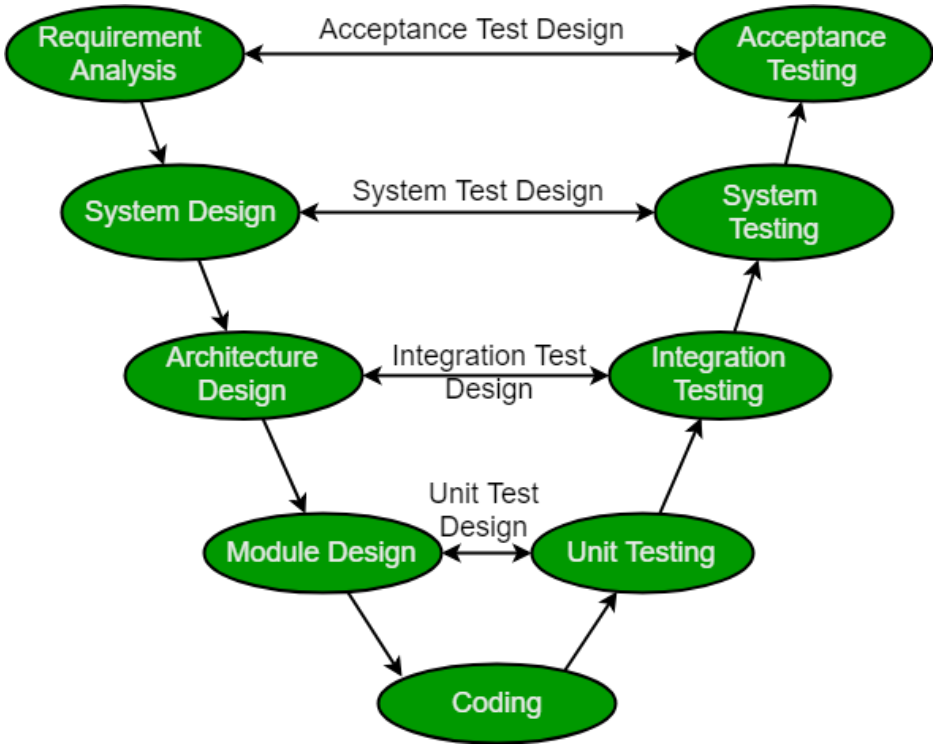


Fig 1.1 V-SHAPED MODEL

Advantages of V-model:

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves alot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

Disadvantages of V-model:

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

Chapter 2

SYSTEM ANALYSIS STUDY

Chapter 2

SYSTEM ANALYSIS STUDY

2.1 INTRODUCTION TO SYSTEM

When we receive a request for a new software project from the customer, first of all, we would like to understand the project. The new project may replace the existing system and sometimes, the new project is an enhancement or extension of a current (manual or automated) system. No matter, whether its functionality is old or new, each project has a purpose, usually expressed in what the system can do. Hence, goal is to understand the requirements of the customer and document them properly. A requirement is a feature of the system or a description of something the system is capable of doing in order to fulfil the system's purpose.

The hardest part of building a software system is declaring precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirements. No other part of the work so cripples the resulting system if done wrong.

No other part is more difficult to rectify later. Throughout software industry's history, we have struggled with this truth .Defining and applying good, complete requirements is hard to work, and success in this endeavour has eluded many of us. Yet, we continue to make progress.

2.1.1PROJECT OVERVIEW

Style transfer

Style transfer is the technique of separating and recombining the content and the style of an arbitrary image. The topic is particularly interesting because it creates artificial intelligence that inter-plays the content and the style of an image to produce artistic results of high quality. Style transfer is the technique of recomposing images in the style

of other images. It all started when Gatys published a paper on how it was actually possible to transfer artistic style from one painting to another picture using convolutional neural networks.

Multi style transfer uses a pre-trained convolution neural network. Then to define a loss function which blends two images seamlessly to create visually appealing art, NST defines the following inputs:

- A content image — the image we want to transfer a style to
- A style image (***s***) — the image we want to transfer the style from
- An input (generated) image — the image that contains the final result (the only trainable variable)

How does it work?

1. We take input image and style images and resize them to equal shapes.

2. We load a pre-trained Convolutional Neural Network (VGG16).

3. Knowing that we can distinguish layers that are responsible for the style (basic shapes, colors etc.) and the ones responsible for the content (image-specific features), we can separate the layers to independently work on the content and style.

4. Then we set our task as an optimization problem where we are going to minimize:

- content loss (distance between the input and output images - we strive to preserve the content)

- style loss (distance between the style and output images - we strive to apply a new style)

- total variation loss (regularization - spatial smoothness to de noise the output image)

Why Neural style transfer?

Deep neural networks have already surpassed human level performance in tasks such as object recognition and detection. However, deep networks were lagging far behind in tasks like generating artistic artefacts having high perceptual quality until recent times. Creating better quality art using machine learning techniques is imperative for reaching human-like capabilities, as well as opens up a new spectrum of possibilities. And with the advancement of computer hardware as well as the proliferation of deep learning, deep learning is right now being used to create art.

Convolution Neural Network

CNNs are some of the most powerful procedures for image processing tasks, and have recently reached human level performance in classification tasks. The neural layers of a CNN can be understood as a set of image filters that extracts higher level features from the pixels of an input image.

CNN is shown to be able to well replicate and optimize the key steps in a unified framework and learn hierarchical representations directly from raw images. If we take a convolutional neural network that has already been trained to recognize objects within images then that network will have developed some internal independent representations of the content and style contained within a given image.

High level architecture

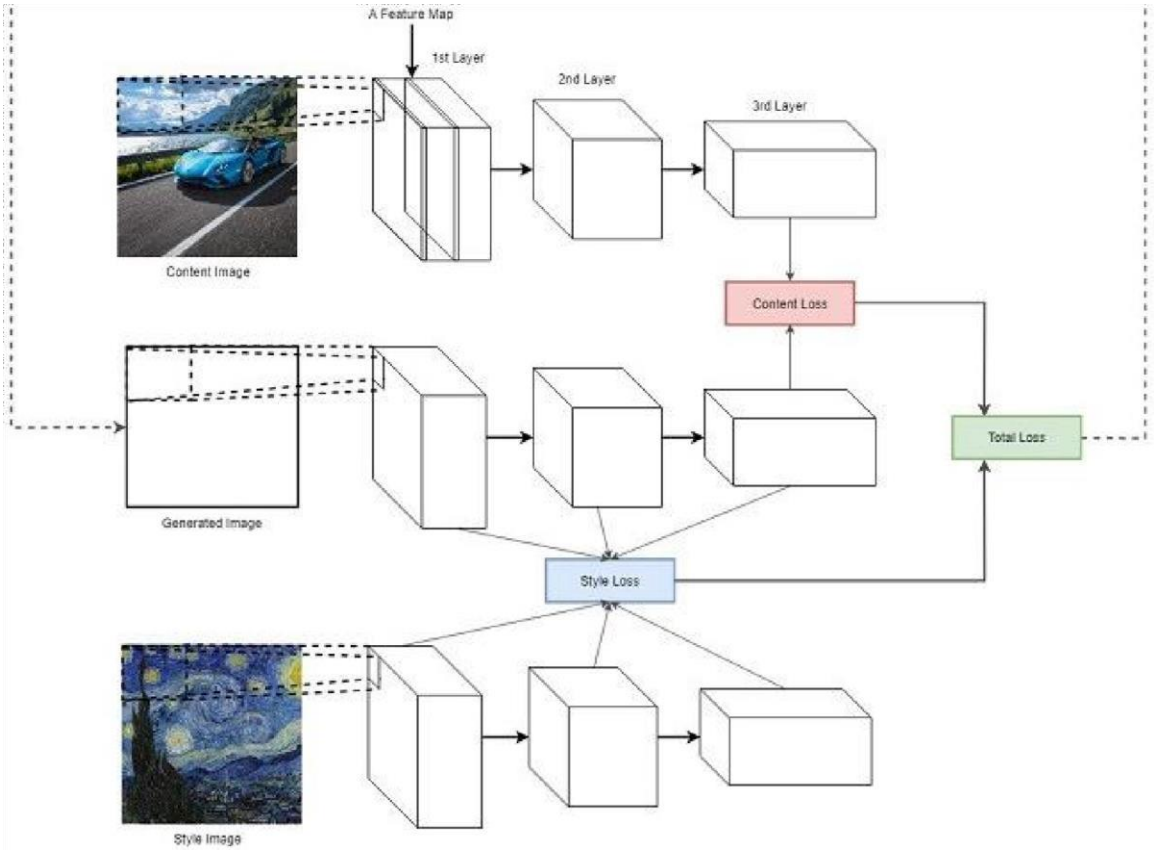


Fig 2.1 high level architecture

Development

Packages used:

Torch: It provides a flexible N-dimensional array or Tensor, which supports basic routines for indexing, slicing, transposing, type-casting, resizing, sharing storage and cloning.

Torch.nn: The nn package is used for building neural networks. It is divided into modular objects that share a common modules interface. Modules have forward() and

backward() method that allows them to feed forward and back propagate, respectively.
Numpy: It supports for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

torch.optim (efficient gradient descents) The optim package defines an Optimizer that will update the weights for us.

PIL, PIL.Image, matplotlib.pyplot (load and display images)

PIL: This library supports many file formats, and provides powerful image processing and graphics capabilities.

PIL.Image: provides a number of factory functions, including functions to load images from files, and to create new images.

Matplotlib.pyplot: You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code.

torchvision.transforms (transform PIL images into tensors)

torchvision.models (train or load pre-trained models)

Addressing different tasks, including: image classification, pixel wise semantic segmentation, object detection, instance segmentation, person key point detection and video classification

copy (to deep copy the models; system package)

Constructs a new compound object and then, recursively, inserts *copies* into it of the objects found in the original.

GPU Detection

We can use **torch.cuda.is_available()** to detect if there is a GPU available. Next, we set the **torch.device** for use throughout the tutorial. If GPU is available then the image size will be 512 otherwise 128.We use small size if there is no GPU present, because it helps in loading the images quickly which are bigger in size or the process will take longer.

Loading Images

We will import the style and content images. The original PIL images have values between 0 and 255, but when transformed into torch tensors, their values are converted to be between

0 and 1. The content and style images also need to be resized to have the same dimensions .

Displaying Images

We create a function that displays an image by reconverting a copy of it to PIL format and displaying the copy using **plt.imshow**. We display the content and style images to ensure they were imported correctly.

Loss Function

In this section we define two loss functions; the content loss function and the style loss function. The content loss function ensures that the activations of the higher layers are similar between the content image and the generated image. The style loss function makes sure that the correlation of activations in all the layers are similar between the style image and the generated image.

Content Loss

Each time the network is fed an input image the content losses will be computed at the desired layers and because of auto grad, all the gradients will be computed. In order to make the content loss layer transparent we define a forward method that computes the content loss and then returns the layer's input. The computed loss is saved as a parameter of the module.

Gram Matrix

A gram matrix is the result of multiplying a given matrix by its transposed matrix. The gram matrix must be normalized by dividing each element by the total number of elements in the matrix. This normalization is to counteract the fact that matrices with a large dimension yield larger values in the Gram matrix during the gradient descent .

Style Loss

The style loss module is implemented similarly to the content loss module. It will act as a transparent layer in a network that computes the style loss of that layer. In order to calculate the style loss, we need to compute the gram matrix. Features tend to be in the deeper layers of the network so this normalization step is crucial.

Now the style loss module looks almost exactly like the content loss module. The style distance is also computed using the mean square error .

2.2 SYSTEM REQUIREMENTS GATHERING

There are different types of requirements such as:

i. **Known requirements** – Something a stakeholder believes to be implemented.

ii.**Unknown requirements**-Forgotten by the stakeholder because they are not needed right now or needed only by another stakeholder.

iii.**Undreamt requirements**-Stakeholder may not be able to think of new requirements due to limited knowledge..

The term stakeholder is used to refer to anyone who may have some direct or indirect influence on the system requirements. Stakeholder includes end-users who will interact with the system and everyone else is an organization who will be affected by it.

A known, unknown, or undreamt requirement may be functional or non-functional.

Functional requirements describe what the software has to do. They are often called product features.

Non-functional requirements are mostly quality requirements that stipulate how well the software does what it has to do. Non-functional quality requirements that are specially important to users include specifications of desired performance, availability, reliability, usability and flexibility. Non-functional requirements for developers are maintainability, portability, testability.

Some requirements are architectural, such as component-naming compatibility, upgradability, etc. Other requirements are constraints, such as system design constraints, standards conformance, legal issues and organizational issues. Constraints can come from users or organizations may be functional or non-functional.

Software requirements are broadly classified as functional and non-functional requirements.

I. **Functional requirements**: These are related to the expectations from the intended software. They describe what the software has to do. They are also called product features.

   Sometimes, functional requirements may also specify what the software should not do.

II. **Non-functional requirements**: Non-functional requirements are mostly quality requirements that stipulate how well the software does what it has to do. Non functional requirements that are especially important to users include specifications of desired performance, availability, reliability, usability and flexibility. Non-functional requirements for developers are maintainability, portability and testability. In the business environment, it is required to have an effective way of market research to understand what a customer wants and how to be successful over competitors. We need to focus on how to make the users to achieve their goals.

The Requirements gathering process will help in understanding the needs of a customer, especially in the IT industry.

There are several different requirement gathering techniques that can be used. Several tools and techniques are used by the stakeholders and business analyst to facilitate this process and capture the exact and detailed requirements. The Requirements gathering techniques should help in breaking down Requirements and Gathering into digestible steps thereby providing instructions to complete each step.

Methods for requirement gathering are as follows:

- Brainstorming
- Prototyping

1. BRAINSTORMING

It involves self-generated contribution of ideas by the group members around a specific issue, problem or requirement. The appropriate subject matter experts will start creatively brainstorming about what the solution might look like. The ideas gathered from the group members will be prioritized depending on the ones they think are the best for this solution. The resulting consensus of best ideas is used for the initial requirements. The objective of brainstorming in a group is to reduce social suppression among group members and stimulate fresh ideas generation leading to an increase the overall creativity of the group.

2. PROTOTYPING

In this approach, the preliminary requirements will be gathered which is used to build an initial version of the solution called a prototype. The prototype may not have all the functionality but serves as a proof of concept for idea verification/further analysis. An iterative process of prototype creation, testing and feedback is followed before reaching a final stage.

2.3 FEASIBILITY STUDY

The feasibility study proposes one or more conceptual solution to the problem set of the project .A feasibility study is a study, usually done by engineers, that establishes whether conditions are right to implement a particular project. Feasibility studies can be done for many purposes, and are sometimes done in IT in order to look at feasibility for new hardware and software setups. Every project is feasible provided given unlimited resources and infinite time. Unfortunately the development of a computer-based system is more likely to be plagued by resource scarcity and stringent schedules. It is both necessary and prudent to evaluate the feasibility of a project at earliest possible time. Wastage of manpower and financial resources and untold professional embarrassment can be avoided if an ill conceived system is recognized early in the development phase.

So a detailed study was carried out to check the workability of the proposed system. Feasibility study is a test of system proposal regarding its workability, impact on the organization, ability to meet user needs and effective use of resources. Thus, when an application is proposed, it is normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis is related in many ways. If project risk is great, the feasibility of producing quality is reduced. There are several types of feasibility.

2.3.1 Technical Feasibility

A study of resource availability that may affect the ability to achieve an acceptable system. Technical feasibility is the most difficult area to ensure at initial stage. Since the objectives, functions, performance cannot be predicted to its fullest, everything seems possible, provided the right assumptions are made. It is essential that the process of analysis and definition can be conducted in parallel with an assessment of technical feasibility. The consideration that is normally associated with technical feasibility includes resource availability at the organization where the project is to be developed and implemented.

2.3.2 Operational Feasibility

It deals with the consideration about working of the system after installation. The proposed system would be beneficial to its users as their needs are fully satisfied. As this project satisfies all the requirements of the users it is operationally feasible. All the operational aspects are considered carefully here. Only by spending tie to evaluate feasibility we will be able to reduce the chances for extreme embracement at later stages of a project.

2.3.3 Financial Feasibility

Financial feasibility should be distinguished from economic feasibility. Financial feasibility involves the capability of the project organization to raise the appropriate funds needed to implement the proposed project. Project financing can be a major obstacle in large multiparty projects because of the level of capital required. Loan availability, credit worthiness, equity, and loan schedule are important aspects of financial feasibility analysis.

There isn't any extra requirement of peripheral or software for development of system as it can be completed with the available resources.

2.3.4 Social Feasibility

Social feasibility addresses the influences that a proposed project may have on the social system in the project environment. The ambient social structure may be such that certain categories of workers may be in short supply or non existent. The effect of the Project on the social status of the project participants must be assessed to ensure compatibility. It should be recognized that workers in certain industries may have certain status symbols within the society.

2.3.5 Market Feasibility

Another concern is market variability and impact on the project. This area should not be confused with the Economic Feasibility. The market needs analysis to view the potential impacts of market demand, competitive activities, etc. and "divertible" market share available. Price war activities by competitors, whether local, regional, national or international, must also be analyzed for early contingency funding and debt service

negotiations during the start-up, ramp-up, and commercial start-up phases of the project. The proposed system is technically, operationally, economically, socially, culturally, markedly feasible.

2.4 SYSTEM REQUIREMENT STUDY REPORT

2.4.1 SYSTEM PROCESS REQUIREMENT

Windows Based Requirements

Computer running Microsoft windows must meet the following minimum hardware and software.

- Dual-core 64-bit processor.
- 8GB of memory.
- Up to 24 GB of internal storage
- Windows 10, Windows 8.1 Update, Windows 8 and Windows.

Technologies Used

- Machine learning
- Python

Application Software

- Google Colab

Chapter-3

SYSTEM DESIGN

Chapter-3

SYSTEM DESIGN

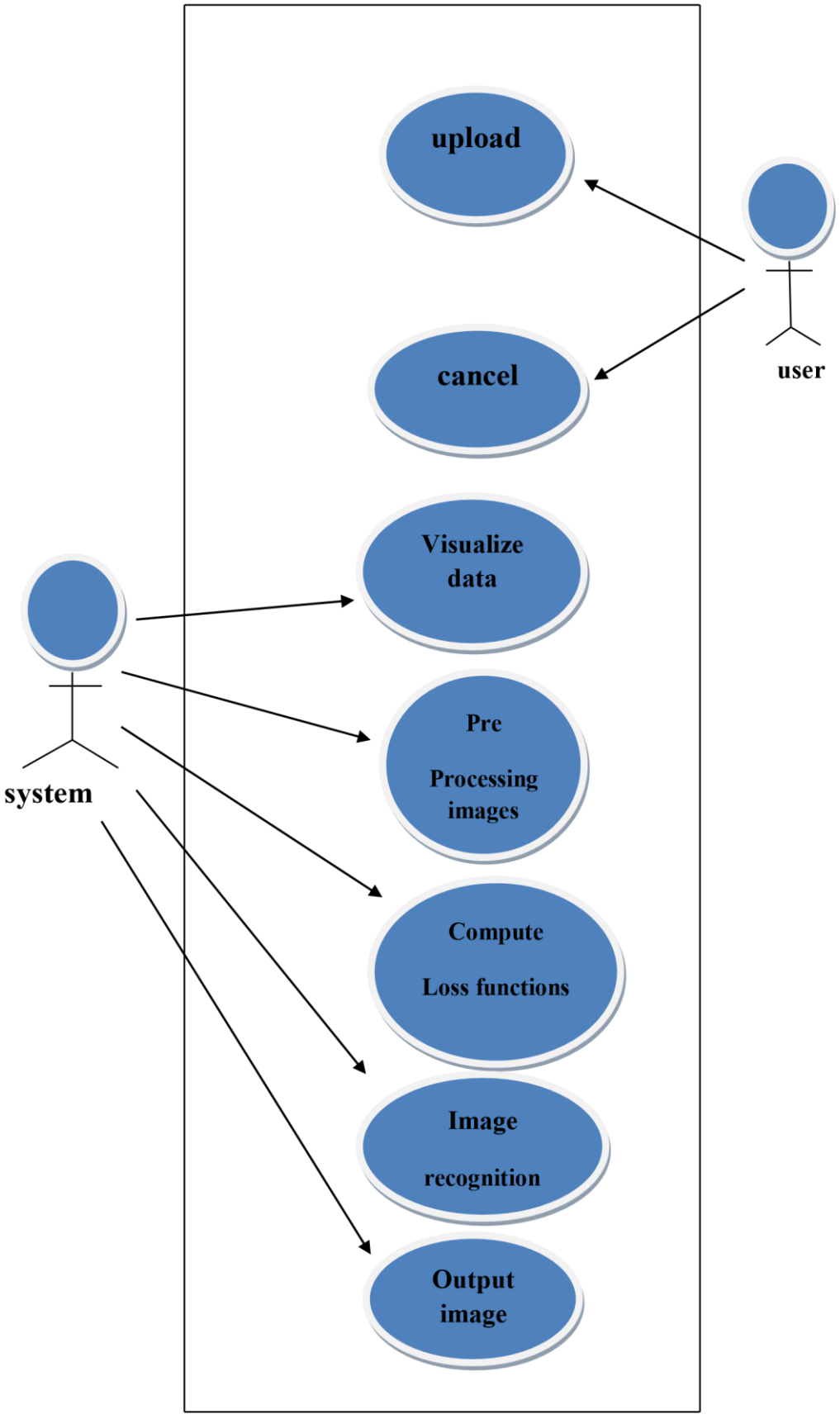
3.1 INTRODUCTION

For declaring that the system is economically feasible, system will be cost effective and budgetary constraints, it should be cheap and quick. There isn't any extra requirement of peripheral or software for development of system as it can be completed with the available resources.
System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

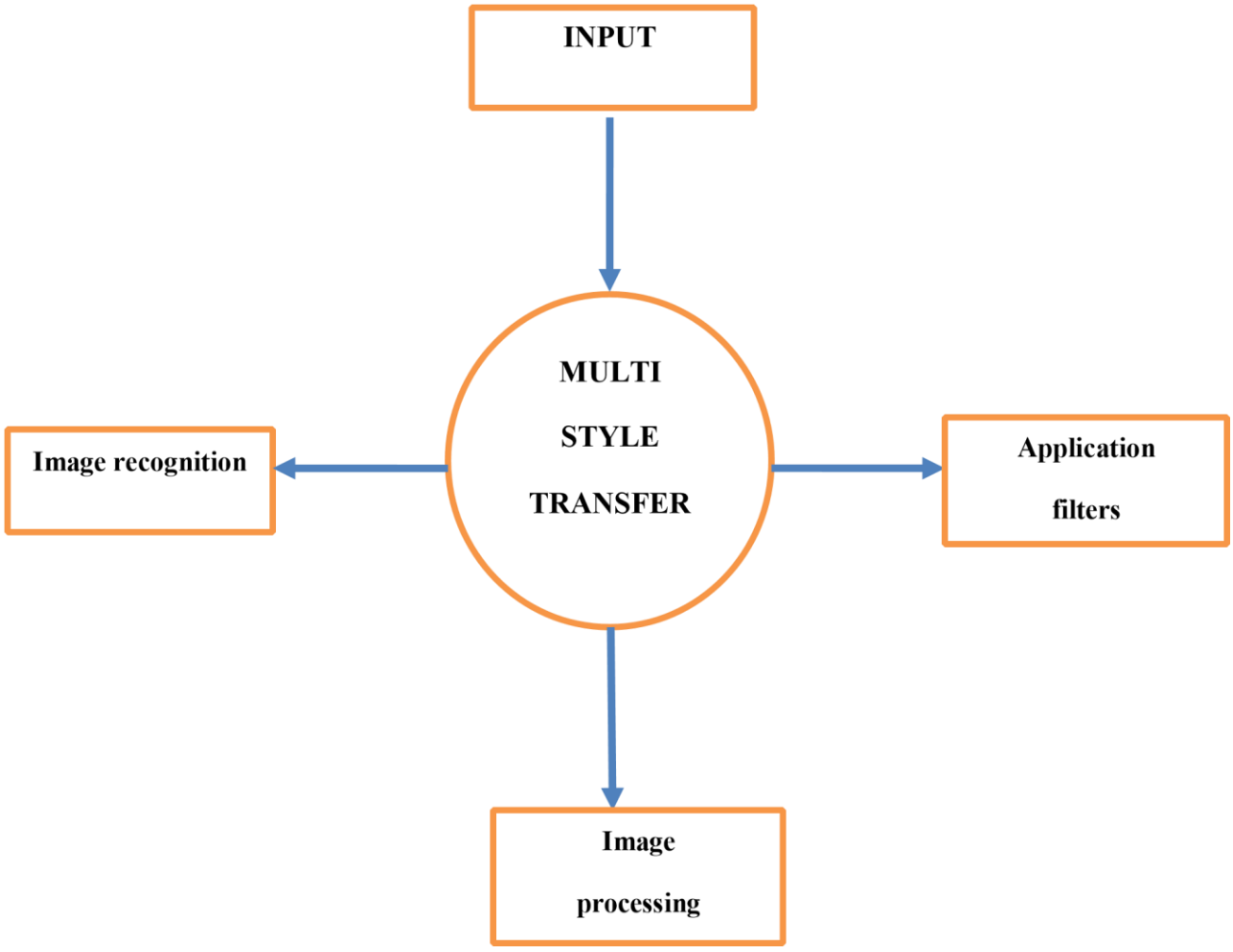
3.2 PHYSICAL DESIGN

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output. It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

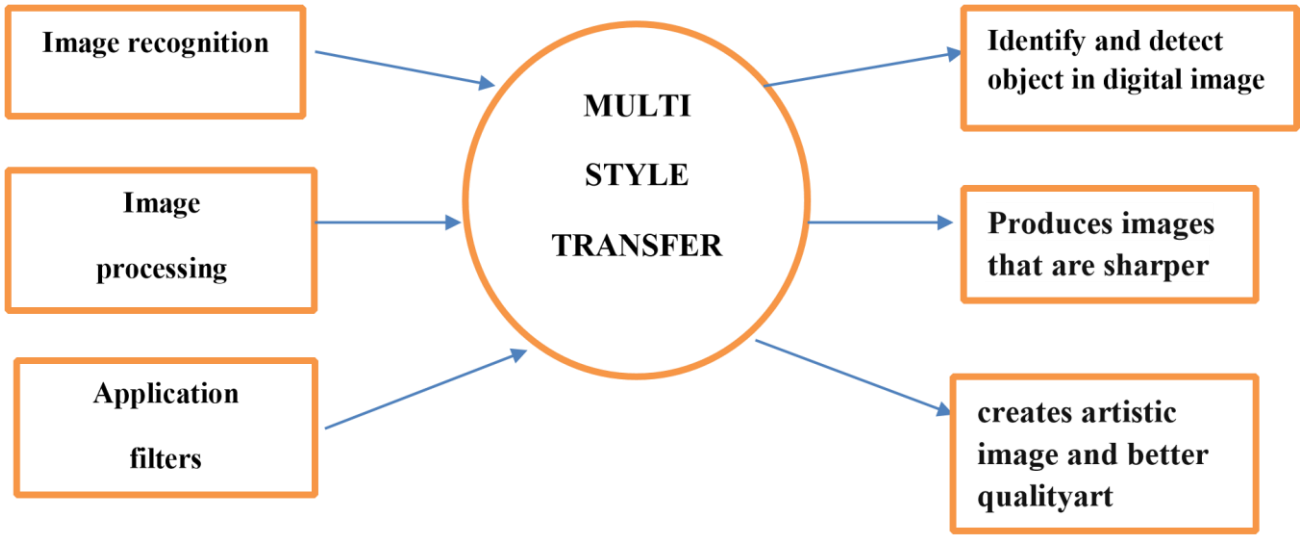
3.2.1 USE CASE DIAGRAM



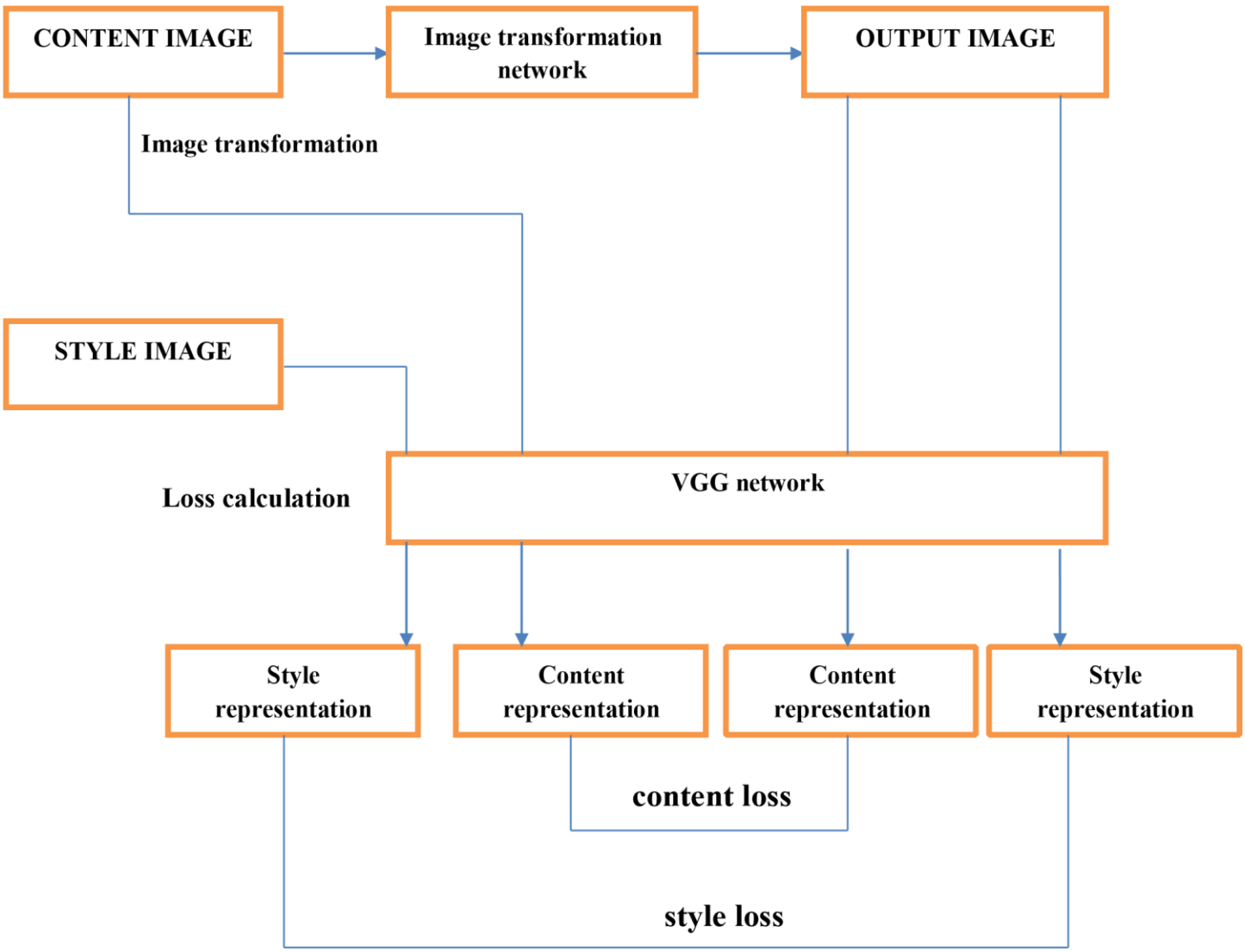
3.2.2 Data Flow Diagram



ZERO LEVEL DFD



FIRST LEVEL DFD



SECOND LEVEL DFD

3.3 USER INTERFACE DESIGN

3.3.1 INPUT SCREEN DESIGNS

Fig 3.1 content image

3.3.2 REPORT LAYOUT DESIGNS

Fig 3.2 output image

Chapter 4

SOFTWARE CODING, TESTING & IMPLEMENTATION

Chapter 4

SOFTWARE CODING, TESTING & IMPLEMENTATION

4.1 INTRODUCTION

Developer tests the software process activities such as design, implementation and requirement engineering. Because, design errors are very costly to repair once system has been started to operate, therefore, it is quite obvious to repair them at early stage of the system, therefore, analysis is the most important process of any project.

4.2 TESTING

Testing is an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a white box and Black box testing.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Acceptance Testing

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit

testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step.

Software often fails at its boundaries.
Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified.

INTEGRATION TESTING

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After unit testing, System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

4.3 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. So, we can divide the activities within the fundamental test process into the following basic steps:

- planning and control
- Analysis and design
- Implementation and execution
- Evaluating exit criteria and reporting
- Test closure activities

1) Planning and Control:

- To determine the scope and risk and identify the objective of testing.
- To implement test policy or test strategy.
- To determine test resources.
- To schedule test analysis and design task, test implementation, execution and evaluation.

2) Analysis and design:

- Review test basis.
- To identify test condition.
- To design the tests.
- To evaluate testability of the requirements and system.
- To design the environment set-up and identify required infrastructure and tools

3) Implementation and execution:

- To develop and prioritize our test cases by using techniques and create test data for those tests.
- To create test suites from the test cases for efficient test execution.
- To execute test suites and individual test cases following the test procedures.
- To re-execute the tests that previously failed in order to confirm a fix. This is known confirmation testing or retesting.
- Comparison of actual and expected result.

4) Evaluation exit criteria and reporting:

- To check the test logs against the exit criteria specified in test planning.
- To assess if more test are needed or if the exit criteria specified should be changed.
- To write a test summary report for stakeholders.

5) Test closure and activities:

- When a project is cancelled.
- When some target is achieved.
- When a maintenance release or update is done.

4.4 TESTCASES

S.NO	TEST SCREEN	PRE-CONDITION	POST-CONDITION	RESULT
1	Loading content image.	Image to be styled must be imported on the system.	Content image is uploaded and displayed.	PASS
2	Loading style images.	Different Style images must be imported.	Style image from which we want style is uploaded and displayed.	PASS
3	Setting up of loss functions.	content losses and style losses will be computed at the desired layers.	Computed loss is saved.	PASS
4	Displaying output image	Third image must display on the system.	Output is generated.	PASS

5	**Changing the style image.**	**Another style image is imported.**	**Different style image is displayed.**	**PASS**
6	**Different output generation.**	**Output with another style is displayed on the system.**	**Output is generated.**	**PASS**

CHAPTER 5

OUTPUT FORMS & REPORTS

CHAPTER 5

OUTPUT FORMS & REPORTS

Fig-5.1 Input image

Fig 5.2 Style image 1

Fig 5.3 Output image`1

Fig 5.4 output image 2 (50% of content image and style image 1)

Fig 5.5 Style image 2

Fig 5.6 Output image 2

Fig 5.7 Style image 3

Fig 5.8 output image 3

CHAPTER-6

CONCLUSION & FUTURE ENHANCEMENT

CHAPTER-6

CONCLUSION & FUTURE ENHANCEMENT

6.1 FEATURES OF STYLE TRANSFER

- Easy to use.
- Style transfer is used for creating artistic fantastic imagery by separating and recombining the image content and style.
- It is very interactive and saves time to change the style of image.
- This approach so far produces images that are sharper than any other generation method.
- One can be artistic and creative using style transfer.
- Creates better quality art using machine learning techniques.

6.2 CONCLUSION

Neural style transfer allows to blend two images (one containing content and one containing style) together to create new art. Style transfer is an optimization technique used to take two images, a *content* image and a *style reference* and blend them together so the output image looks like the content image, but painted in the style of the style reference image. This is implemented by optimizing the output image to match the content statistics of the content image and the style statistics of the style reference image. It creates artificial intelligence that inter-plays the content and the style of an image to produce artistic results of high quality.

We will follow the general steps to perform style transfer:

1. Visualize data
2. Basic Preprocessing/preparing our data

3. Set up loss functions
4. Create model
5. Optimize for loss function

6.3 FUTURE SCOPE

- The system can be developed in such a way that its existing features can be modified to better versions.
- Deep learning and neural networks are undoubtedly going to fundamentally change the way we create media and art of all kinds.
- Future work involves training the network on a large number of styles and experimenting with different architectures of the style prediction network and different methods of integrating the style prediction network into the feed-forward network.
- Much progress can be made on style transfer to extend the style transfer technique from still images to videos and even audio and other mediums.

BIBLIOGRAPHY

- L.A. Gatys, A.S. Ecker, and M. Bethge, Image Style Transfer Using Convolutional Neural Networks, CVPR, 2016.

REFERENCES

- https://web.stanford.edu/class/ee368/Project_Autumn_1617/Reports/report_wang_tan.pdf
- http://cs231n.stanford.edu/reports/2017/pdfs/405.pdf
- www.youtube.com