**SUPERIOR UNIVERSITY LAHORE**

|  |
| --- |
|  |

**Faculty of Computer Science & IT**

**FINAL YEAR PROJECT**

**PROJECT PROPOSAL & PLAN**

**[NuraStyle]**

Project ID: **[Issued by FYP Manager]**

**Project Team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Program** | **Contact Number** | **Email Address** |
| Usama Arif | BCSM-F16-060 | BSCS | 03478147119 | bcsm-f16-060@superior.edu.pk |
| Mehwish Aslam | BCSM-F16-440 | BSCS | 0323 0045619 | bcsm-f16-440@superior.edu.pk |
| SYED MURTAZA ABBAS | BCSM-S17-027 | BSCS | 0305 4029994 | bcsm-s17-027@superior.edu.pk |
|  |  |  |  |  |
|  |  |  |  |  |

**[Project Supervisor]**

([Designation])

**[NuraStyle]**

**Change Record**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author(s)** | **Version** | **Date** | **Notes** | **Supervisor’s Signature** |
|  | 1.0 |  | Original Draft |  |
|  |  |  | Changes Based on Feedback from Supervisor |  |
|  |  |  | Changes Based on Feedback From Faculty |  |
|  |  |  | Added Project Plan |  |
|  |  |  | Changes Based on Feedback from Supervisor |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Project Proposal**

**Project Title:** [14 pt, Calibri]

# Executive Summary

[12 pt, Calibri, Justified]

[*An executive summary summarizes a longer report or proposal or a group of related reports in such a way that readers can rapidly become acquainted with a large body of material without having to read it all.* *This section summarizes the overall document, and should include the important highlights from the document. It should be concise. It is NOT an introduction, index or table of contents, it is a summary. The Executive Summary should not make any reference to other parts of the document. You have to write one page to let reader understand an overview of the project.]*

NuraStyle is a mobile app that transfers the style of an image into an art. A Mobilenet Convolutional neural network Trensorflow model would be converted to TFlite and optimized to run on mobile devise without needing to upload image into datacenter or needing to use any external Nvidia GPU. This ensures the privacy of the user as well as rendering image without incurring a round trip to the server. The key feature that makes this app unique would be introducing users to select their own art style image from gallery, as previously users had to stay bound to use styles available by the developers.

# Introduction

[12 pt, Calibri, Justified]

[*Between 5 to 10 lines that briefly introduce users for which solution is being developed. It gives the relevance of problem and gives background information to reader*]

Nurastyle intends to provide style transferring of an image into and art without the need of a cloud server. For this TensorFlow Lite model would be implemented in an android device to efficiently work without having to wait a lot for the processing. A pretrained CNN model would be Re-trained on PC for later converting it into a TensorFlow lite model. During the conversion, we’ll compress model by reducing weights and integer values down to 8-bit float from straight 32-bit floating value.

1. **Existing System / Competitive Analysis**

[12 pt, Calibri, Justified]

[*Between 15 to 20 lines that describe existing system and current situation. Write the previous related work by others, literature review with credible sources, and list down all the possible competitors of your product. That is, you need to list down all those products that are closely related to your product in terms of features, target audience, etc.*]

Not late after tensorflow was first introduced in 2015, German researchers came up with style transfer technique which was soon after seen in many applications released on mobile devices in 2016 starting from Prisma and later Artisto and Faceapp. Everyone an image had to process was to be sent to servers first. With slowly improvement in Tensorflow, Google introduced Tensorflow lite

There are many applications that introduced style transfer technique that used CUDA compute power so uploading image has been essential. We have seen huge growth in applications related to AI as it makes easier for users to simply apply a filter without having to manually do it. Faceapp, Prisma, Artisto and some other applications have provided similar features but users are concerned about their privacy as images are uploaded to developer’s

1. **Problem Statement**

[12 pt, Calibri, Justified]

[*Between 5 to 10 lines describing problem that motivates for new solution. The problem statement should be concise, as you now understand it.*]

Relying on an internet connection.

Privacy Issues that are raised when a user intends to use those app.

Previously apps consumed many seconds to render an image.

And most importantly, users had to rely on filters available in the application and couldnt manually insert it by themselves.

## Proposed Solution

[12 pt, Calibri, Justified]

[*Between 10 to 20 lines in which solution of the problem will be proposed.*]

Simple solution would be to use Tensorflow model to be converted into tensorflow lite so the whole model is compressed which would be able to run on mobile devices allowing users to simply run whole model on their mobile device without needing to even connect to internet. Whole model would be trained on a computer directly on a Nvidia GPU with CUDA Compute power and compressed on desktop machine too. Once the model is accurate and weights are reduces, an android app would simply convert the image into 512px so the image which would be the required size for an input image. Same goes for the Art.

There would be 2 images, source and destination image. Model would try to match the corresponding style and content target for an input image.

Android phones are enough capable to run AI smoothly now and in future they will be getting more powerful in terms of computation, so the only 512px image would take only a second to transfer style of the image.

1. **Scope of the Project**

[12 pt, Calibri, Justified]

[*Major modules / features will be discussed in this section. Each module will be discussed in 5 to 6 lines.**Present the features that are included and the possible features that are excluded from your scope.*]

**Latency**: You don’t need to send a request over a network connection and wait for a response. This can be critical for video filtration applications that process successive frames coming from a camera all in real-time.

**Availability**: The application runs even when outside of network coverage as it solely would utilize mobile phone’s computing power.

**Speed**: New hardware specific to neural networks processing provide significantly faster computation than with general-use CPU alone. The model would be optimized enough to provide result in around a second for even Mid end phones with Snapdragon 600-700 series.

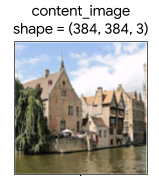
**Privacy**: The data does not leave the device.

**Cost**: No server farm is needed when all the computations are performed on the device.

1. **System Architectural Design**

Trained TensorFlow Model

TensorFlow Lite converter



TensorFlow Lite model



Style\_image shape=(256,256,3)

Android App

JAVA AP



Interpreter (Kernels)

Android Neural   
Network API

1. **Implementation Tools and Techniques**

[12 pt, Calibri, Justified]

[*Describe your methodology for implementation along with implementation tools*.]

We’ll consider using mobilenet pretrained model that is lighter and has a good enough accuracy.

TensorFlow lite has a converter where you ingest the saved model to save it into the.tflite file. So we’ll go for converting it and later pretrain it. There is an interpreter for actually executing the interface. It’s called TensorFlow Lite dialect. The use of interpreter will be to allocate minimal load on memory, and to ensure execution latency.

Tensorflow interface to follow these steps:

1. **Loading a model**

We’ll be loading .tflite model into memory which contains model’s weights.

1. **Transforming data**

We’ll need to resize the image to match the weights, such as converting image file format and its resolution. To make it compatible with the model.

1. **Running inference**

This step involves using the TensorFlow Lite API to execute the model. It involves a few steps such as building the interpreter, and allocating tensors, as described in the following sections.

1. **Interpreting output**

As we receive the results from the model inference, we need to interpret outputs and convert it into an image.

**Hardware:**

The networks were trained on a computational device with the following hardware specifica-tions.

• GPU: Nvidia GTX 1050 Ti (4GB)

• CPU: Intel Xeon E3-1240-42 @ 3.6 GHz

• RAM: 16GB 1600MHz

1. **Project Plan**

[12 pt, Calibri, Justified]

[*This section describes how the project will be managed, including a detailed plan with milestones. Project Plan includes two things i.e. Work breakdown structure (WBS) and Gantt Chart consisting of Timeline with milestone.*]

* 1. **Work Breakdown Structure**

[*A work breakdown structure (WBS) is deliverable based decomposition of project scope. The WBS includes 100% of the work defined by the project scope and captures all deliverables – internal, external, interim – in terms of the work to be completed, including project management. You have to give division of responsibilities and duties among team members*.]

# Roles & Responsibility Matrix:

The purpose of roles & responsibility matrix is to identify who will do what.

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #** | **WBS Deliverable**  **Activity** | **Definition of Activity or Task (Description)** | **Responsible Team Member(s) & Role(s)** |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Tensorflow Convertion |  |  |
| 1.1 | Hardware | Platform where training need to done. | Usama Arif |
| 1.2 | Testing pretrained model | Understanding model archetecture | Usama Arif |
| 1.3 | Train Model | training model using our datasets | Usama Arif |
| 1.4 | tensorflow lite converter | converting into .tflite file format | Usama Arif |
|  | Control Flow & Training on-device | Support for training on-device, focused on personalization and transfer learning | Usama Arif |
| 2 | Optimization |  |  |
| 2.1 | Post Training Quantization | Quantization for 8-bits fixed-point to reduce size and latency. | Usama Arif |
| 2.2 | During-training quantization | Quantization for 8-bits fixed-point lowering bits precision | Usama Arif |
|  | Re-training model | Re-training with different weights to aquire our target model size | Usama Arif |
| 3 | Android Studio Integration |  | Mehwish Aslam |
| 3.1 | Front End design | Design layout | Mehwish Aslam |
| 3.2 | Android optimization | For app to run smoothly on multiple android versions and resoloution. | Mehwish Aslam |
| 3.3 | Embedding TF lite model | Drag & drop TFLite models into Android Studio | Mehwish Aslam |
| 3.4 | Tensorflow Lite support | Implement interprerter for TfLite | Usama Arif |
|  |  |  |  |
|  |  |  |  |
| 4 | Performance |  | Syed Murtaza Abbas |
| 4.1 | Better tooling | Public dashboard for tracking performance gains with each release | Syed Murtaza Abbas |
| 4.2 | GPU backend optimizations | implement vulkan support for low level api to utilize gpu efficiently | Usama Arif |
| 4.3 | model performance | Measuring and evaluating model performance | Syed Murtaza Abbas |
| 4.4 | Evaluating model inference time | Deploying on mobile device to check overall time taken | Syed Murtaza Abbas |
|  |  |  |  |
| 5 | Documentation & Presentation |  | Whole Group |

* 1. **Gantt Chart**

[*Gantt Chart consisting of Timeline with milestone. Specific items to include in this section are as follows:*

* *Timeline with milestones:* ***Gantt chart in Microsoft project 2013.*** *The following are required elements of your Gantt chart:*
* *Project duration is from the date your project is enrolled to the completion date:*
* *Each milestone is to be labeled with a title.*
* *Schedule all tasks not just “Design” or “Testing.” Break this schedule down to specific assignments.*
* *Each task is to be labeled with a title and person or persons assigned to the task.*
* *Subdivide larger items so that no task is longer than about one week*
* *Link tasks which are dependent on the completion of a previous task.*]

**References**

Mobile Object Detection using TensorFlow Lite and Transfer Learning | Oscar Alsing

Tensorflow Blog

Google Developers

Nvidia Developers

Image Style Transfer Using Convolutional Neural Networks | University of Tubingen, Germany

Neural Style Transfer: A Review | IEEE

MobileNetV2: Inverted Residuals and Linear Bottlenecks

Transfer Learning using Mobilenet and Keras | Terhat Culfaz

**List of Faculty Proposed Changes**

**Project Title**

|  |  |  |
| --- | --- | --- |
| **Proposed Change** | **Proposed By** | **Supervisor’s Decision** |
|  | Name of Faculty Member(s) who proposed this change | Approved/Disapproved and/or Comments |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Supervisor’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**APPROVAL**

|  |  |
| --- | --- |
| **Project Supervisor** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |



|  |  |
| --- | --- |
| **Project Manager** | |
| Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |