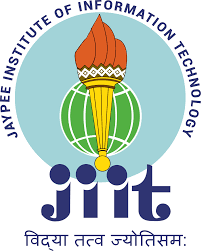
horizontal line

****

**MINOR PROJECT SYNOPSIS**

**FONT PAIRING**

Team Members: - Submitted to: -

1. Siddhartha Singhal – 16803010 Mrs. Sakshi Agarwal
2. Aadish Garg – 16803020
3. Prakhar Vaish – 16803025

Font Pairing

**Minor Project Report**

# TEAM MEMBERS (B13) SUBMITTED TO:

* Aadish Garg 16803020 Mrs. Sakshi Agarwal
* Prakhar Vaish 16803025
* Siddhartha Singhal 16803010

# 

# INTRODUCTION

Font pairing is a classic problem in the design world. Different fonts can be used to draw attention, lead the eye, or even form the foundations of a brand identity.  
Pairing fonts that are very similar, but just slightly different creates visual conflict. This is actually a core part of design - contrast is important not only in font selection but color and position as well.  
  
  
  
A common way to combine fonts is to use fonts in the same family, or created by the same designer. Another approach is to match various typographic measures, like x-height and ascenders/descenders.  
Font pairs that are both far from each other and oriented vertically/horizontally are better candidates, because they share one dimension in common.  
Since fonts vary by a lot more than just obliqueness and weight, we have to add more dimensions. Eg. A Z-axis for serifs vs sans-serifs.

# KEY FEATURES

* Presetting user preferences on the basis of a psychological criterion
* Initializing a model based on the inference of user preference
* A model that clusters different font attributes like font weights, font family, contrast between fonts, obliqueness, x-height and ascension.
* User friendly UI to generate page templates and suggestions
* Using real world dataset from google font

# LITERATURE SURVEY:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source** | **Publisher** | **Algorithm Used** | **Datasets used** | **Merits of methods** | **Demerits of methods used** |
| http://cs231n.github.io/transfer-learning/ | Dr. K.Karpathy, Stanform University | Transfer Learning | ImageNet(1.2M Size) | Reduces training time, Re-use already trained weights | None |
| github.com/Jack000/fontjoy | Handle : Jack000 | None | FontJoy Font Dataset | Pretrained font vectors, Font images dataset | None |

# MODULE FLOW CHART

# 

# MODULE DESCRIPTION

|  |  |  |
| --- | --- | --- |
| **Module/Function Name** | **Algorithm/ Pseudocode** | **Description** |
| Dataset processing | Pytorch, sklearn, img2vec | These libraries are used for text preprocessing and vectors of length 512 are generated. |
| Training module | Keras, Optimizer: SGD, Error function: mean squared error | Neural network that trains the model on predefined dataset. It establishes the relationship between different font. 65 fonts are used to train the neural net. |
| Cosine Similarity | Sklearn metrics | It takes font as an input and returns a font that is most similar to the input font |
| UI | Bootstrap, HTML, CSS | Simple and clean UI |
| API | Flask API | UI interacts with api and passes a font as input. It passes the input font to neural net and predicts an output and passes this to cosine similarity. Most similar font is returned to the user and is embedded using google CDN. |
| Dataset Plotting | Pyplot and matplotlib | We used two open source libraries to plot the dataset. Data set contains 200 vectors and we used PCA for dimensionality reduction and reduced it to 3 dimensions. Then, pyplot and matplotlib are used. Matplotlib is slower in comparison to pyplot. |

# 

# RESULTS:

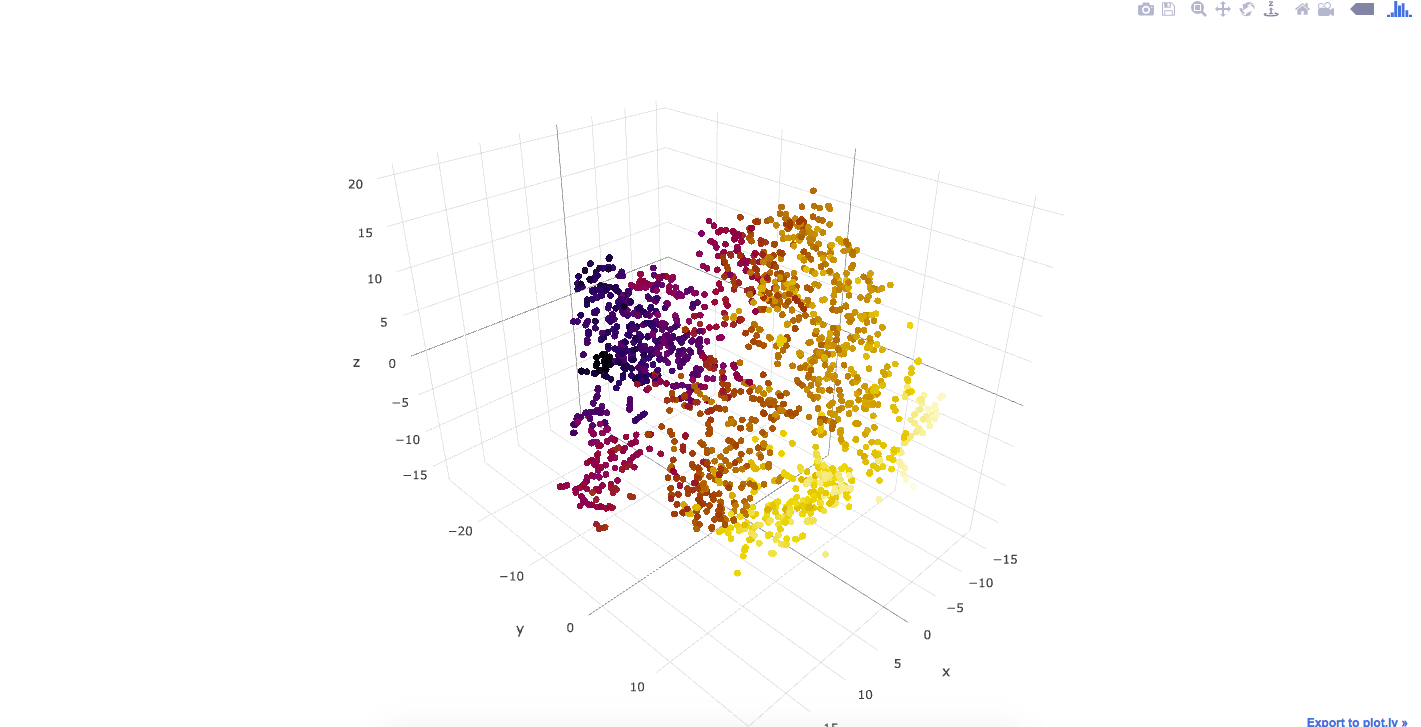
**a) Datasets Used:**

1. **FontJoy Font Dataset :** [**https://github.com/Jack000/fontjoy/blob/master/fonts-vectors200.json**](https://github.com/Jack000/fontjoy/blob/master/fonts-vectors200.json)
2. **Paired-Fonts-Datasets :** [**https://kaggle.com**](https://kaggle.com)

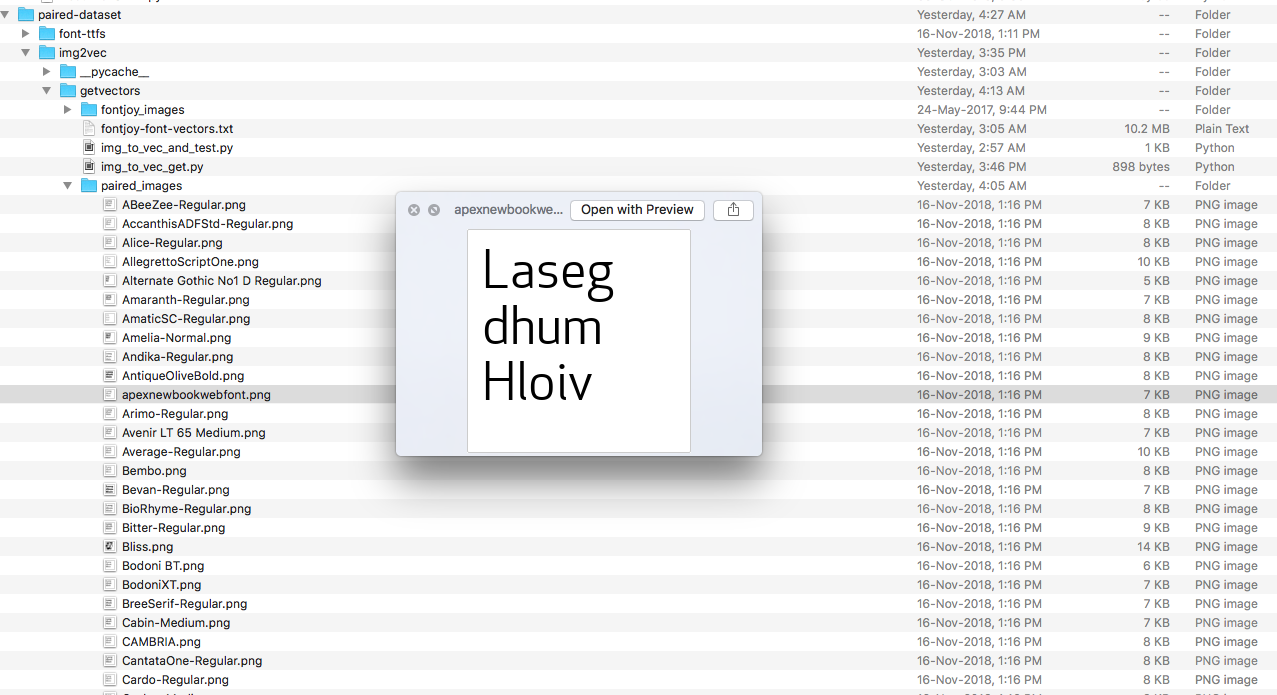
**b) Hardware/Software Requirements :**

**Hardware :** Core i5/ 8 GB RAM  
 **Software :** Mac OS/ Linux environment with listed libraries in modules

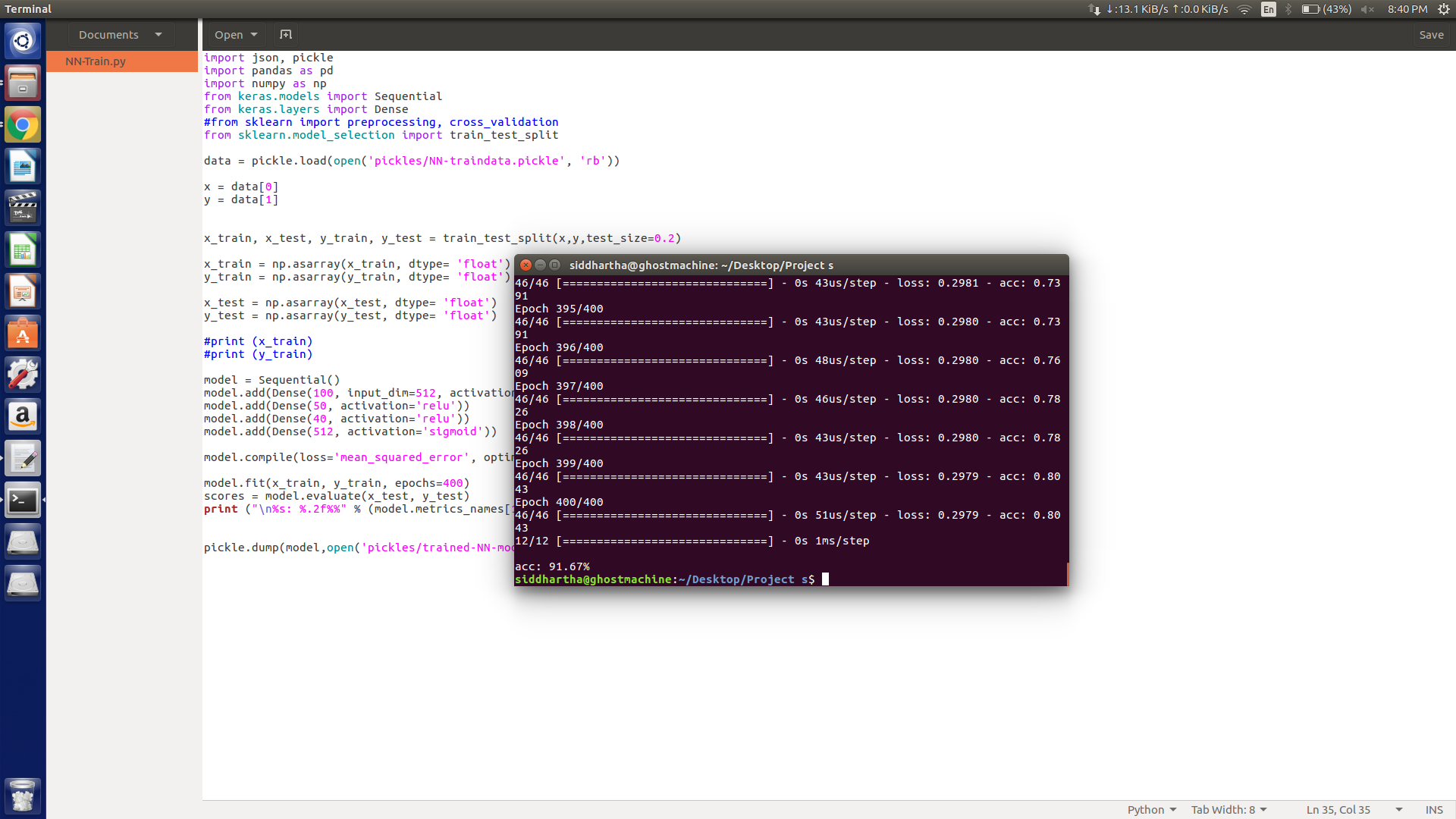
**c) ScreenShot Table :**

Visualization of FontJoy Dataset :   


Generated Images from paired-font dataset :



Trained Neural Network. Accuracy shown.



# 

# SUSTAINABLE GOALS ACHIEVED

1. Quality Education – The application of different fonts for the use making educational presentation interesting is one of the key goals of this project.
2. Decent Work and Economic Growth – Automating the selection and design of page layout to ease the works of the designers.

# CONCLUSION

We achieved 90% accuracy through training a neural network and developed relationship between various attributes of heading and the paragraph content. Developed an application that allows users to easily interact with the backend. The API written in flask predicts and outputs the matching font allowing users to change their preferences.

# REFERENCES

1. http://cs231n.github.io/transfer-learning/
2. https://github.com/tensorflow/tensorflow
3. http://blog.ethanrosenthal.com/2016/12/05/recasketch-keras/