Face clustering and texture Classification based fast image searching.

NOMADIUM

~Adheesh Trivedi and Aditya Sinha

The project aims to address the common challenge of navigating through directories containing a large collection of images and videos, enabling users to efficiently filter and search for their own or others' images.

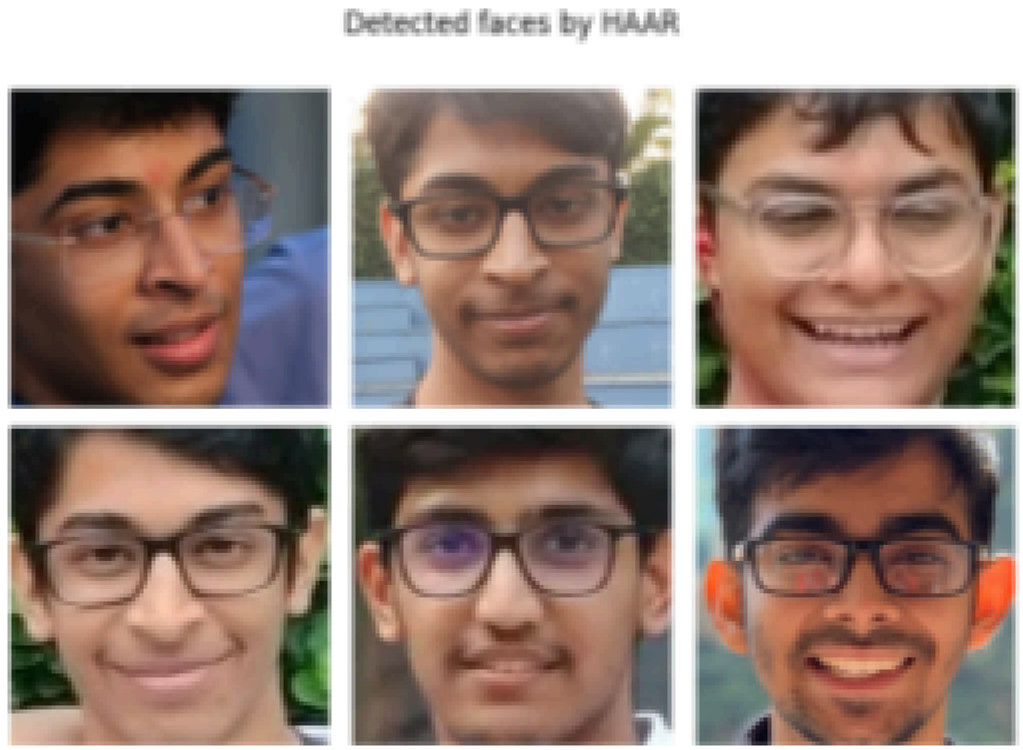
To achieve this, the solution focuses on two key objectives: **face clustering** and **texture-based classification**. The initial phase of the project implements these functionalities to simplify and enhance the search process in large datasets.

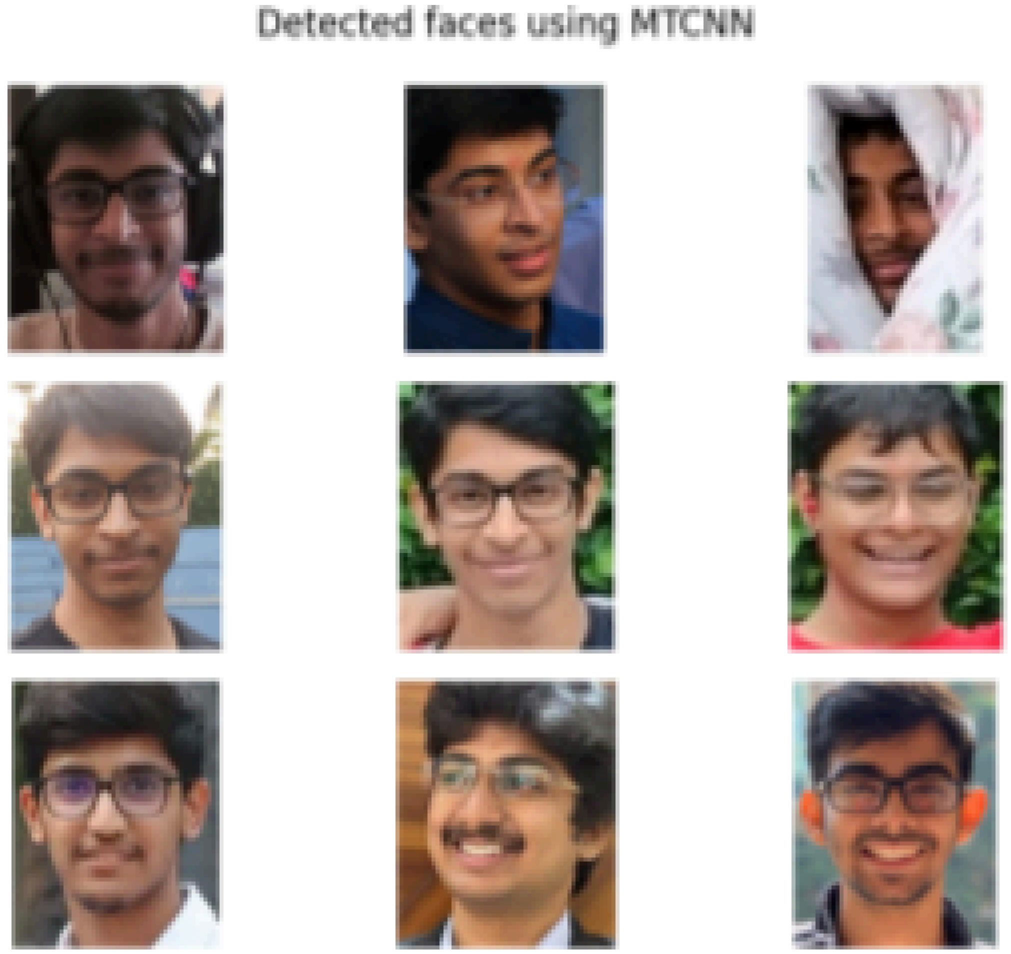
- **Face Clustering**: This module, developed by Adheesh, categorizes images based on the faces they contain, grouping similar faces together to facilitate streamlined browsing.
- **Texture Classification**: A comparison based study of different feature descriptors with different ML and DL models to find out the best feature descriptor and what all parameters does the classification depends on.

These components form the foundation of the project's vision to assist users in effectively managing and exploring extensive multimedia collections.

Face detection

- Our first job is to detect the faces. We tested Viola-Jones[1] and MTCNN[2] face detectors.
- Viola-Jones is Fast, but not robust to rotation and occlusion. Thus we go for the DL approach of MTCNN.
- MTCNN was made to run on GPU and works in real time.





A bit on how both the approaches work and why one is superior to the other.

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features,"
[2] L. Zhang, H. Wang and Z. Chen, "A Multi-task Cascaded Algorithm with Optimized Convolution Neural Network for Face Detection,"

Face clustering and texture based fast image searching.

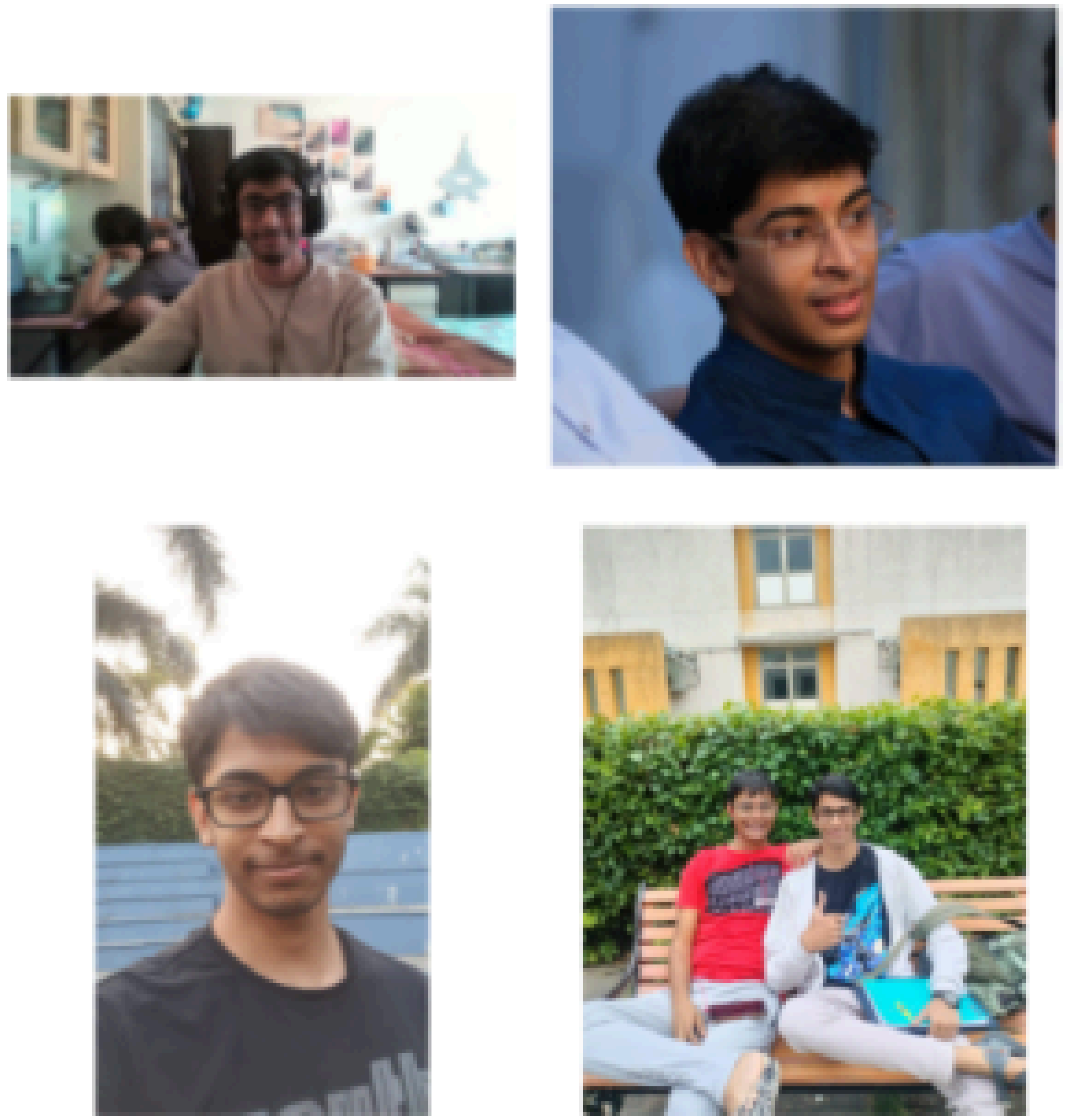
NOMADIUM

~Adheesh Trivedi and Aditya Sinha

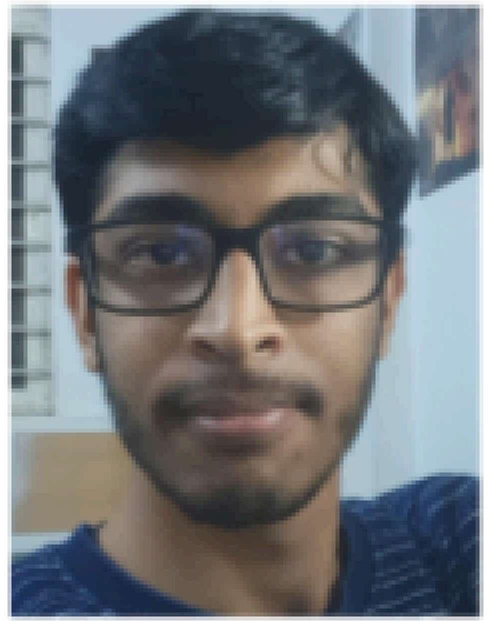
Face identification

- Using Inception Resnet-V1[1] pretrained on VGGFace2[2][3] dataset.
- Images are batched and sent to GPU for applying the model. VRAM is considered while batching.
- Faces are matched using “Norm distance” and a threshold of .8 to 1.2

Matched images



Template used for matching



[1] F. Schroff, D. Kalenichenko, J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering,"
[2] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman. “VGGFace2: A dataset for recognising face across pose and age, International Conference on Automatic Face and Gesture Recognition,”
[3] Pretrained model taken from https://github.com/timesler/facenet-pytorch/

NOMADIUM

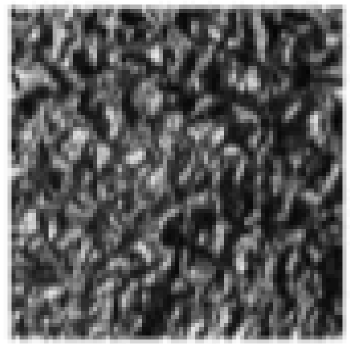
~Adheesh Trivedi and Aditya Sinha

Texture Classification: Comparison based study

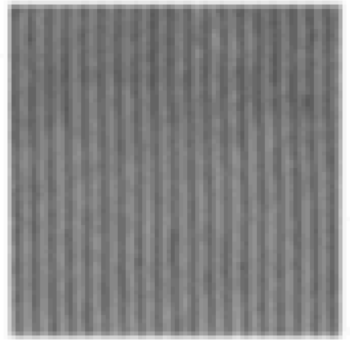
AIM: To compare different feature descriptors for texture classification using machine learning and deep learning models.

Dataset Description:

- KTH-TIPS (TEXTURE UNDER VARYING ILLUMINATION PATTERNS) Dataset
- 3 Major Classes (Crumpled Aluminium Foil, Orange Peel, Corduroy)
- 81 images each classes
- Camera used: OLYMPUS C-3030 zoom
- Images taken at 9 different scales
- Size of images (200*200) , Gray Scale Images



Crumpled aluminium foil



Corduroy

Orange peel

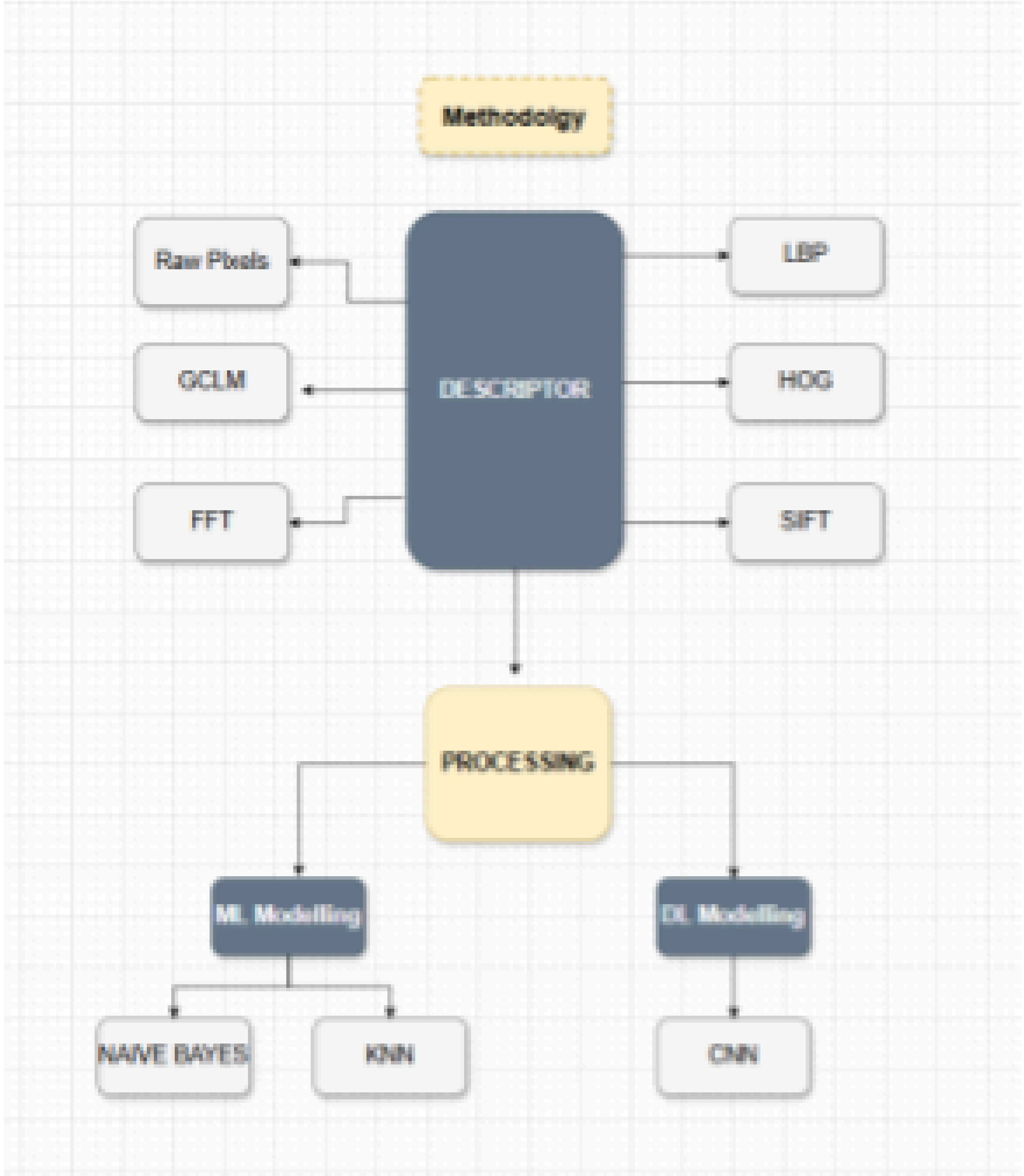
Face clustering and texture based fast image searching.

NOMADIUM

~Adheesh Trivedi and Aditya Sinha

Process Flow Diagram

- 6 Different feature descriptors are considered for input
- Each of these features passed as an input separately to the 3 selected models.
- For the input we have considered a 50-50 train test split.
- In order to process this data we have considered mainly 3 models **Naive Bayes, K-Nearest Neighbour and Convolutional Neural Network**
- In order to evaluate the model we have used the **Macro Averaged F1-score**



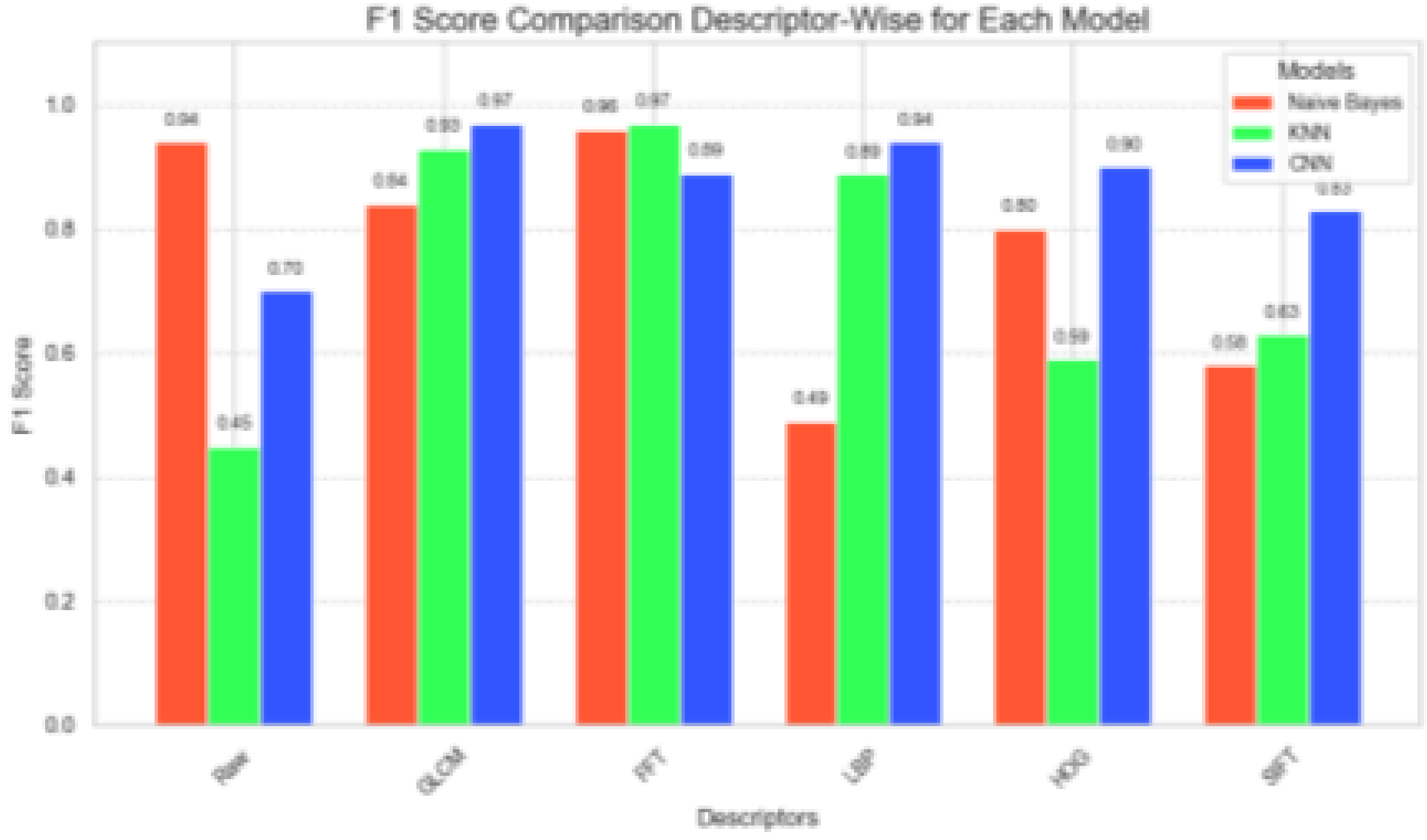
Face clustering and texture based fast image searching.

NOMADIUM

~Adheesh Trivedi and Aditya Sinha

Results

- GlCM & FFT were most consistent through models.
- Naive Bayes surprisingly performed well on raw pixel.
- LBP requires complex model to get utilized fully.
- SIFT performing bad continuously shows that it is not a good texture descriptor.



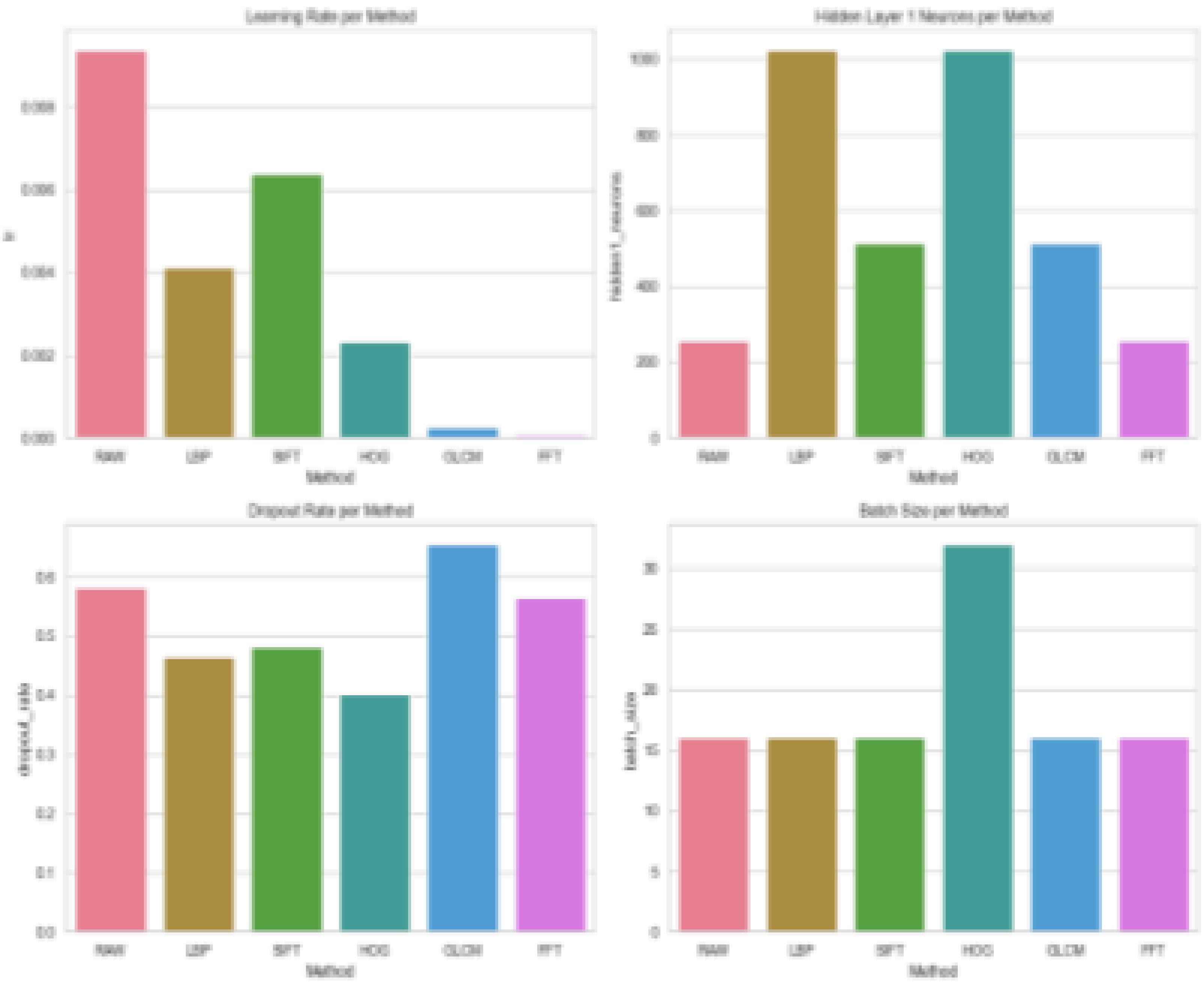
Face clustering and texture based fast image searching.

NOMADIUM

~Adheesh Trivedi and Aditya Sinha

HYPER-PARAMETER TUNING OF CNN

- Two fully connected layers has been used and the and the outputs decided by hyper parameter tuning.
- At each layer ReLU activation function is used that adds non-linearity to the model.
- Drop outs has been decided using hyper parameter tuning.
- Dropout prevents overfitting.
- Loss function used is Cross Entropy Loss



Conclusions

- Naive Bayes performing well on Simple models shows its assumptions of iid data.
- KNN performing bad on High Dimensional Data.
- CNN is the best model to have texture classification.

Limitations

- In a comparison-based model, we can't use every model because not all models are suited for every type of data or problem. Some models might be too slow, too complex, or require too much data to work properly.
- High computational demand.
- Concept drift.