# Introduction

Human emotion detection is implemented in multiple domains for capturing information about the person and for security purpose. It is an extension of face recognition, which may add as another layer of security, where along with face, emotions and expressions are predicted. This emotion detection is used in business promotions. Most of the businesses thrive on customer responses to all their products and offers, and based on emotions or expression of user image, it allows to make a decision on whether the customer liked or disliked the product or offer. ML approaches considered to make this emotion classification model are K Nearest Neighbour (KNN) and non-linear Support Vector Machine (non-linear SVM using RBF Kernel) algorithms. These two were chosen to compare performance of unsupervised model over supervised model, and had higher accuracy compared to other algorithms in their respective categories.

The unsupervised algorithm- KNN was chosen because it uses the similarity between instances to make predictions. It does not make assumptions about the underlying data distribution and can handle non-linear decision boundaries. Each instance in the training data is represented by it's feature values and corresponding class label, and during the training phase, KNN stores the training instances in memory. When a prediction is required for a new instance, KNN searches for the K nearest neighbour in the feature space using distance metric- Manhattan distance. The class label of the new instance is then determined based on the majority vote or weighted voting of its K nearest neighbours.

Unlike linear classifiers such as linear SVM, KNN does not explicitly learn a linear decision boundary. Instead, it makes predictions based on the local structure of the training data. Thus, it can handle non-linear decision boundaries and is suitable for multi-class classification problems. So KNN is a simple yet effective algorithm for emotion classification in images. It classifies new images based on their similarity in visual features of emotions to that of the known images in the training data. It can handle non-linear relationships between features and emotions since it doesn't make any assumptions about the data distribution, and by selecting an appropriate value for K, KNN can capture the local structure of the data and make accurate predictions for emotion classification.

The supervised algorithm- non-linear SVM with RBF (Radial Basis Function) kernel was chosen because the RBF kernel is a non-linear kernel that can map the input data into a higher-dimensional feature space, where it becomes possible to find a hyperplane that separates the data. This allows SVM to handle complex, non-linear relationships between input features and class labels, thereby capturing non-linear patterns and relationships for better classification. By using the RBF kernel, SVM can effectively model non-linear relationships between features and emotions. It aims to find an optimal hyperplane that maximally separates the different emotion classes in the feature space. The RBF kernel helps SVM to map the input features into a higher-dimensional space where non-linear relationships can be captured, helps handle complex decision boundaries, and can generalize well to unseen images.

# Methodology

Both ML models are classification models which use CKPLUS image dataset from Kaggle for training using labelled data, and use images from Natural Human Face Images for Emotion Recognition in Kaggle as unlabelled testing dataset. But there is imbalance in classes of CKPLUS dataset with some classes having less than 100 images, some having more than 200 images, and the others having between 100 to 200 images. These models can classify a total of 7 emotions- anger, contempt, disgust, fear, happy, sadness, and surprise. The images in both datasets are in grayscale, and have size of 48x48 and 224x224 pixels respectively. The training dataset has 981 images and the testing dataset has 70 unlabelled images extracted from the above mentioned Kaggle dataset, 10 images for each emotion. When training dataset is made to have 80-20 split for train and validation dataset, each have 784 and 197 images respectively.

## I. KNN Model

Pre-processing- The class labels are mapped to numeric labels since it is easier to work with numeric data than categorical data, and are stored in a NumPy array. Then using OpenCV library in Python, we iterate through the images in each folder and convert them into grayscale (to identify distinct structural features for different emotions in image), resize to 48x48 pixels (to ensure all images are of same size), normalize image to float datatype and store it in a list which is later converted to a NumPy array.

Feature extraction- The HOG (Histogram of Oriented Gradients) method is used for extracting features from processed images. This technique counts occurrences of gradient orientation in the localized portion of an image, and the HOG descriptor focuses on the structure or the shape of an object. It is better than any edge descriptor as it uses magnitude as well as angle of the gradient to compute the features. For the regions of the image it generates histograms using the magnitude and orientations of the gradient. We get a total of 900 features extracted from the images which is stored in a NumPy array.

Training- We train the model using features extracted by HOG above. As mentioned earlier, we perform a 80-20 split on the training dataset to split it into train and validation dataset having labelled data. KNN algorithm is applied on train dataset with K value as 1 and distance metric as Manhattan distance. Then we predict class labels (emotions) for images in validation dataset.

Evaluating- We evaluate the model by finding accuracy of labels predicted for validation dataset by comparing predicted label with actual label of image. Cross validation is performed by splitting it into 5 parts, and average score of cross validation is found along with accuracy for each emotion in validation dataset.

Testing- The unlabelled test dataset is used for testing the model by selecting 8 random images from it and predicting their labels (before that, it has to be pre-processed and have it's features extracted in the same way as training dataset). Since it is unlabelled, accuracy can't be found the way we found with validation dataset.

II. Non-linear SVM Model

Pre-processing- Using OpenCV library in Python, we iterate through the images in each folder and resize them to 48x48 pixels (to ensure all images are of same size) and store it in a list which is later converted to a NumPy array.

Feature extraction- The HOG (Histogram of Oriented Gradients) method is used for extracting features from processed images. We convert the resized images to grayscale and store the features extracted from these images in a NumPy array.

Training- We train the model using features extracted by HOG above. As mentioned earlier, we perform a 80-20 split on the training dataset to split it into train and validation dataset having labelled data. Non-linear SVM classifier algorithm using RBF kernel is applied on train dataset with C (regularization parameter) value as 1. Then we predict class labels (emotions) for images in validation dataset.

Evaluating- We evaluate the model by finding accuracy of labels predicted for validation dataset by comparing predicted label with actual label of image. Cross validation is performed by splitting it into 5 parts, and average score of cross validation is found along with accuracy for each emotion in validation dataset.

Testing- The unlabelled test dataset is used for testing the model by selecting 8 random images from it and predicting their labels (before that, it has to be pre-processed and have it's features extracted in the same way as training dataset). Since it is unlabelled, accuracy can't be found the way we found with validation dataset.

# Result

I. KNN Model

KNN model gave an accuracy of 100% when predicting for validation dataset-

```
Shape of X_train: (784, 900)
Shape of y_train: (784,)
Shape of X_test: (197, 900)
Shape of y_test: (197,)
Accuracy: 1.0
Cross-validation scores: [1.         0.95918367 0.93877551 0.98469388 0.9744898 ]
Average score: 0.9714285714285715
Accuracy for anger: 1.0
Accuracy for contempt: 1.0
Accuracy for disgust: 1.0
Accuracy for fear: 1.0
Accuracy for happy: 1.0
Accuracy for sadness: 1.0
Accuracy for surprise: 1.0
```

But when predicting for unlabelled test dataset, the predicted labels didn't match their corresponding emotion for many of the images, and so the model has a poor performance.

Predicted Emotion: anger


Predicted Emotion: disgust

For example, in the above outputs we can see that fear and sadness are wrongly predicted as anger and disgust. One reason might be because in training dataset, the images labelled anger have the feature of wide protruding eyes (angry eyes) and images labelled disgust have the feature of creases near mouth in face with the mouth making "ew" expression, similar to the wide eyes of the child with fear and similar to the expression in mouth region for baby in other image respectively. Another reason for incorrect prediction might be overfitting of model, as it was noted that each image in training (CKPLUS) dataset was duplicated thrice and only images of adults belonging to a particular continent/descent in the world were present, and hence proper generalization for unseen data might not have been possible here. Some predictions from unlabelled dataset which were correct-

Predicted Emotion: surprise

Predicted Emotion: happy

In feature extraction, both HOG and LBP (Local Binary Pattern) methods were applied separately to the model, and since HOG method gave slightly better accuracy, LBP method was discarded. Similarly, Manhattan distance metric gave slightly better accuracy compared to other distance metrics like Euclidian and Hamming, so Manhattan distance was chosen.

Accuracy for this model is highest when K value is 1 and size of image is 48x48 pixels. When K value increases, accuracy of both validation and test dataset decreases, so generalization didn't occur. When size of image (in pixels) was decreased, number of features extracted became less and it decreased accuracy. Increasing image size more than it's original size to extract more number of features also decreased the accuracy. But when increasing number of extracted features without altering original size of image by tuning parameters of HOG method like orientations, pixels per cell, etc., it didn't affect accuracy. So it can be assumed that increasing number of extracted features may not always lead to improved accuracy due to additional features not having relevant or discriminative information for better classification.

It was also noted that this model isn't suitable for bigger datasets with variety of people like Asian, American, etc., as accuracy when tested with validation dataset itself was very low (approx. 39%) for such datasets, resulting in underfitting. Reducing size of training dataset set also led to underfitting, so accuracy would drastically drop if we removed one or more emotion image dataset (like removing entire images of anger from dataset) completely from training (CKPLUS) dataset.

## II. <u>Non-linear SVM Model</u>

This model has an accuracy of approximately 98% for validation dataset-

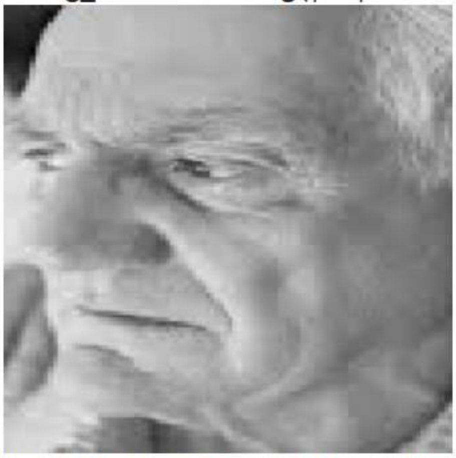
```
Overall Accuracy: 0.9796954314720813
Cross-validation scores: [0.97969543 0.96428571 0.96938776 0.9744898  0.98979592]
Average score: 0.97553092302911
Accuracy for anger: 1.0
Accuracy for contempt: 0.875
Accuracy for disgust: 1.0
Accuracy for fear: 0.9333333333333333
Accuracy for happy: 1.0
Accuracy for sadness: 0.875
Accuracy for surprise: 1.0
```

If we keep value of C (regularization parameter) greater than 1 when using RBF kernel for SVM classifier, we get an accuracy of 100% in validation dataset, but it results in overfitting and class labels (emotions) for unlabelled dataset is not at all predicted correctly. When C value is less than 1 (like C=0.8, 0.01, … since range of C is (0 , infinity)), then also accuracy decreases as underfitting occurs. But when C=1, the model predicts better than KNN model for unlabelled dataset, thus indicating a better generalization for unseen data compared to KNN model. Some of the correct predictions are-

Predicted Emotion: ['disgust']

Predicted Emotion: ['sadness']

For unlabelled test dataset, this model was able to predict emotions correctly for at least half of the images in it. The remaining incorrect predictions might be caused by lack of proper training using more images, and more generalization may be required. Some of the incorrect predictions are shown below. Here, contempt and sadness are predicted incorrectly as surprise and anger respectively-



Predicted Emotion: ['surprise']

Predicted Emotion: ['anger']

In feature extraction, both HOG and LBP (Local Binary Pattern) methods were applied separately to the model, and since HOG method gave slightly better accuracy, LBP method was discarded. When size of image (in pixels) in training dataset was decreased, number of features extracted became less and it decreased accuracy. Increasing image size more than it's original size to extract more number of features also decreased the accuracy.

It was also found that this model is unsuitable for larger datasets with many images as accuracy for both validation and test dataset were very less (approx. 49%). Reducing size of training dataset set also led to underfitting, so accuracy would drastically drop if we removed one or more emotion image dataset (like removing entire images of anger from dataset) completely from training (CKPLUS) dataset.

## Conclusion

So in conclusion, it was found that non-linear SVM model using RBF kernel is better than KNN model for image classification by predicting emotions. This may be because the RBF kernel allows SVM model to learn non-linear decision boundaries between different emotions, thus capturing more complex relationships between features and resulting in better discrimination between different emotions.

Unlike KNN models, non-linear SVM models with RBF kernel are good at handling noise or variations in the image data like lighting conditions, facial expressions, or image quality due to their ability to find support vectors and focus on relevant instances to reduce impact of noisy data on classification accuracy. In terms of generalization, SVM models with RBF kernel can generalize well to unseen data, whereas KNN models may have limitations in generalization, especially when the dataset size increases.

The challenges faced were tuning hyperparameters while applying classification algorithms and finding appropriate feature extraction methods such that classification accuracy is good. To improve performance of KNN model, some other feature extraction methods, image augmentation, usage of other distance metrics, etc. can be tried. Similarly to improve SVM classifier model, different feature extraction methods, image pre-processing methods/augmentation, different gamma values, etc. can be tried. In general, the model has to be trained on bigger dataset with images of people under different age groups (children, adults, old people), race/continent (Asian, African, etc.), gender, facial features (tattoos, scars, moustache, etc.), different angles of face, etc. Hyperparameters like regularization parameter and gamma value in SVM model, and number of clusters (K value), distance metric, etc. in KNN model have to be tuned such that underfitting and overfitting doesn't occur. Only then we'll get a good classification model with high accuracy.

## References


HOG code reference- https://www.thepythoncode.com/article/hog-feature-extraction-in-python

https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Dataset- https://www.kaggle.com/datasets/shawon10/ckplus

Dataset from which unlabelled data was taken for testing- https://www.kaggle.com/datasets/sudarshanvaidya/random-images-for-face-emotion-recognition

Importing dataset directly to Colab from Kaggle- https://www.geeksforgeeks.org/how-to-import-kaggle-datasets-directly-into-google-colab/


## Appendix

There are 2 files named Samuela_Assignment ML_Emotion_classification_KNN.ipynb and Samuela_Assignment ML_Emotion_classification_SVM.ipynb respectively for KNN and SVM models. These are run in Google Colab and require kaggle.json file to be uploaded along with granting permission to Colab for accessing Google drive having test folder for working of both models.