UNIVERSITY OF
ILLINOIS CHICAGO

College of Business Administration

IDS 576 Deep Learning and Applications (41099), Fall 2023

PROJECT REPORT ON

Understanding Facial Expression through People's response

Group- 3

Urvashiben Veerabhai Patel: 656803288

Tanmay Rewari: 656245314

Vinay Khandelwal: 679706443

PROBLEM STATEMENT

Facial expressions serve as crucial non-verbal cues in communication, conveying emotions through images in image processing [1]. Recognizing these expressions holds significance in interpersonal communication, akin to voice tone. Machines interpreting human emotions can have wide-ranging benefits, from aiding healthcare with therapeutic robots to assisting those with disabilities [3]. Emotion recognition can simplify expressing desires and has various applications like retail analysis, crime scene assessment, educational program assessment, and enhancing human-computer interactions [2][4].

Current facial expression recognition applications span emotion analysis, cognitive science, virtual reality, and human-computer interaction [10]. This project explores using CNN and DeepFace approaches to create a reliable classifier for human emotion in both static images and real-time webcam feeds [5][6]. We aimed to classify emotions, including happiness, sadness, anger, disgust, fear, surprise, and neutrality according to Ekman's emotional categories. We investigated the impact of image qualities, such as Gaussian blur, and Gaussian noise, on emotion prediction. This analysis was carried out using manipulated images derived from our dataset, and various deep learning models, including CNN and DeepFace, were employed to assess the performance under these image manipulations. The goal was to understand how these factors influence the models' ability to accurately predict emotions in facial expressions.

INTRODUCTION

Facial expression recognition involves human or computer-based techniques. Our project focuses on two main steps:

1. **Facial Feature Extraction**: This step involves recognizing faces in each environment and extracting their key features, like detecting face shapes and components (e.g., eyes). For instance, we used Haar-cascade in OpenCV for this purpose.

2. **Facial Expression Interpretation**: Here, we analyze the movement or changes in facial features and categorize this data into various emotional or attitudinal categories. We trained three different models for this:

- **CNN Model**: Convolutional Neural Networks, tailored for processing grid-like data like images. These typically consist of convolutional, pooling, and fully connected layers [7].

- **DeepFace**: A Python module for face recognition and analysis of facial attributes like age and gender, encompassing the latest AI facial recognition models [8]. This is free and Open-source library.

3. **Image Manipulation:** Additionally, we experimented with image manipulation using Python with incorporated Gaussian Noise and Gaussian Blur techniques and observe their impact on the CNN and DeepFace models' ability to predict emotions [19].

Lastly, we aimed to predict emotions in real-time through a webcam feed using the CNN model and DeepFace along with face annotation.

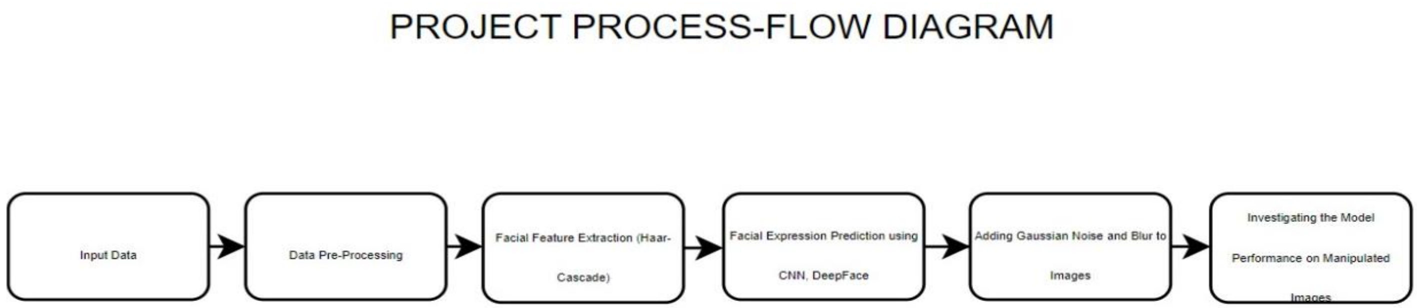


Figure [1]: Problem formulation of our project

DATASET

The Kaggle Facial Expression Recognition Challenge (FER2013) [11] provided the data set that we utilized to train our models. The columns in the dataset are:

- Emotion: The emotion variable represents emotions.
- Pixels: Pixel's variable expresses the value per pixel in the photos
- Usage: Usage shows which set the row it belongs to such as training and testing.

Anger, disgust, fear, happiness, sadness, surprise, and neutral are the seven emotions represented in the dataset's 48 x 48-pixel grayscale pictures. Each face is assigned to one of seven categories, with 0 denoting anger, 1 denoting disgust, 2 denoting fear, 3 denoting happiness, 4 denoting sadness, 5 denoting surprise, and 6 denoting neutralities.

Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise
4953	547	5121	8989	6198	6077	4002

Figure [2]: Number of images for people's expressions from FER2013 dataset

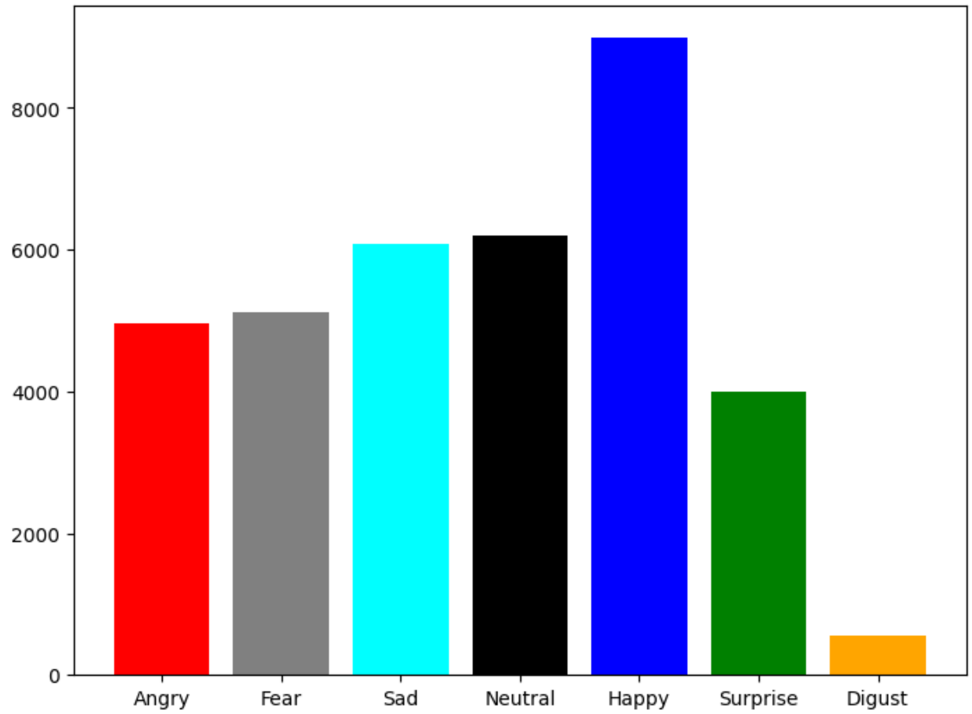


Figure [3]: Class Distribution of the dataset

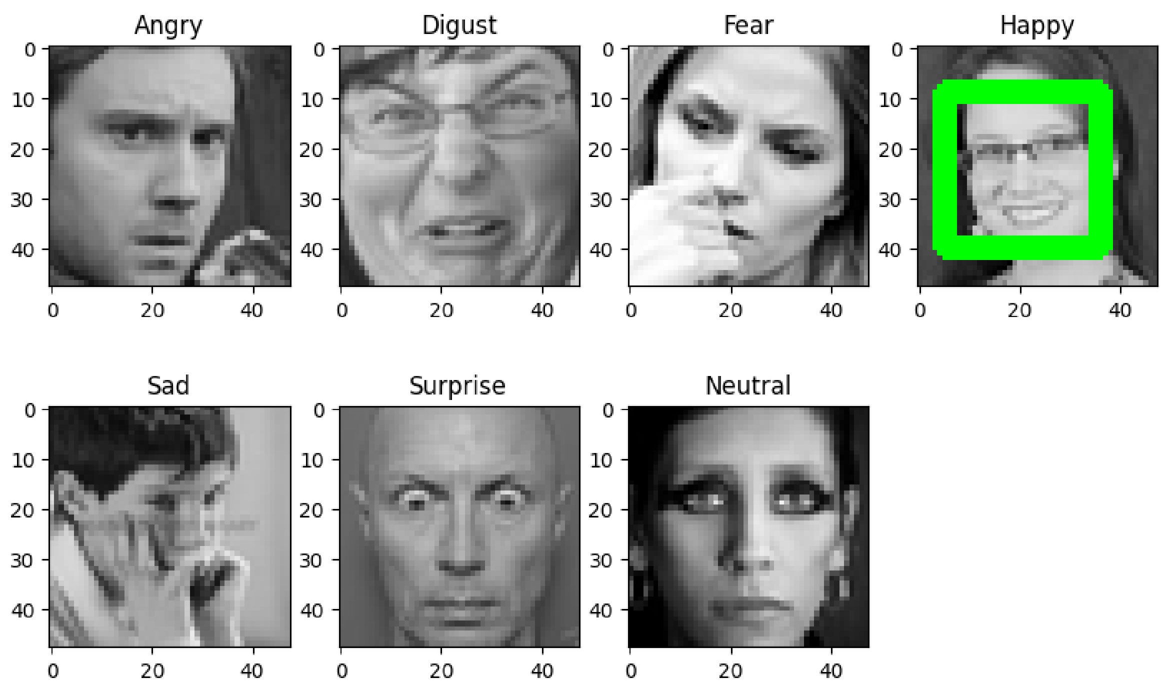
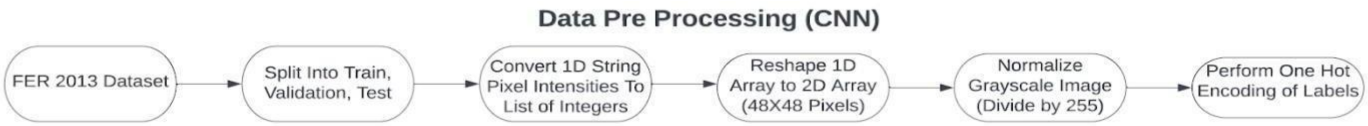
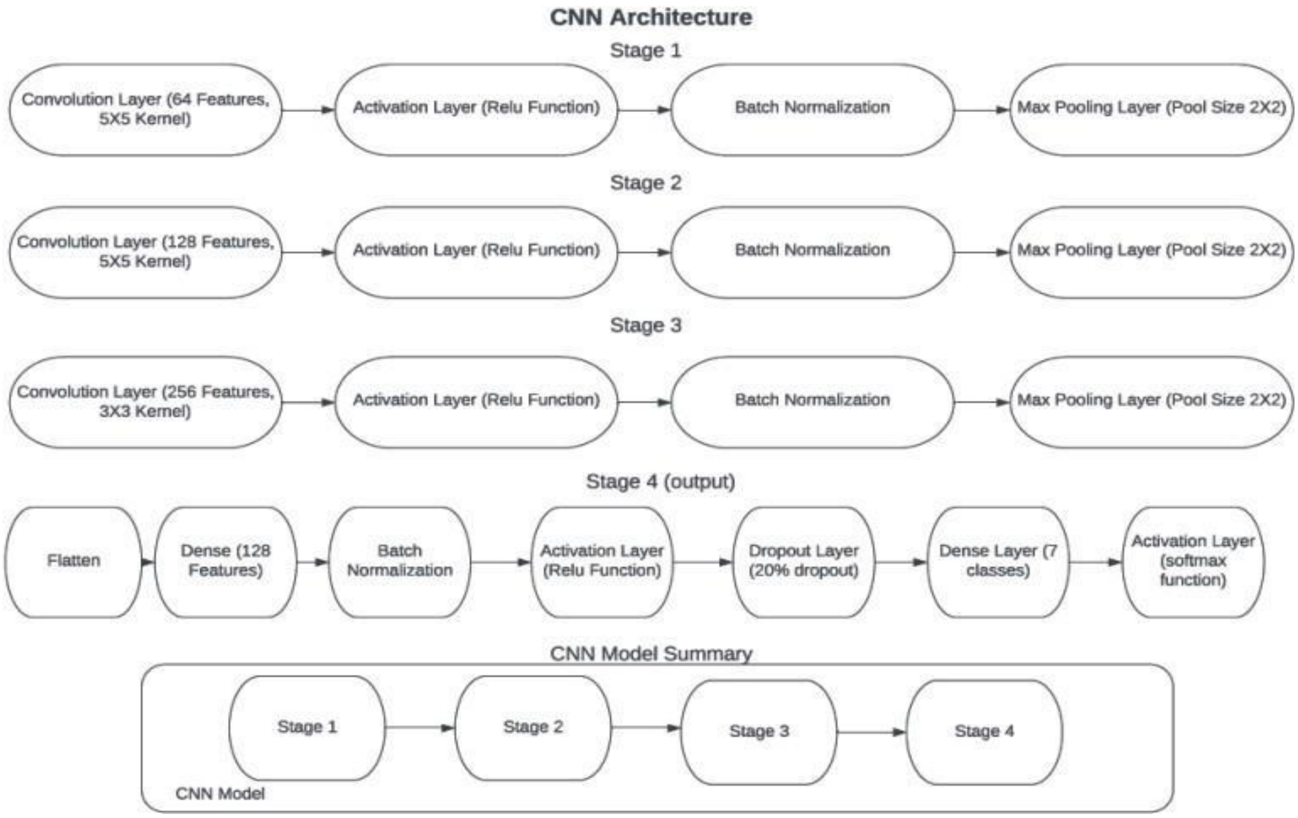
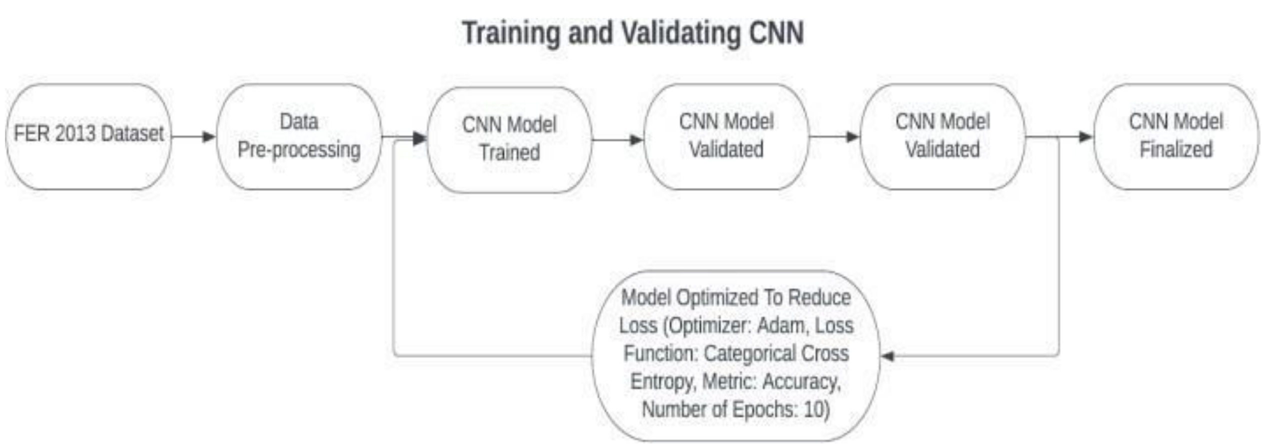


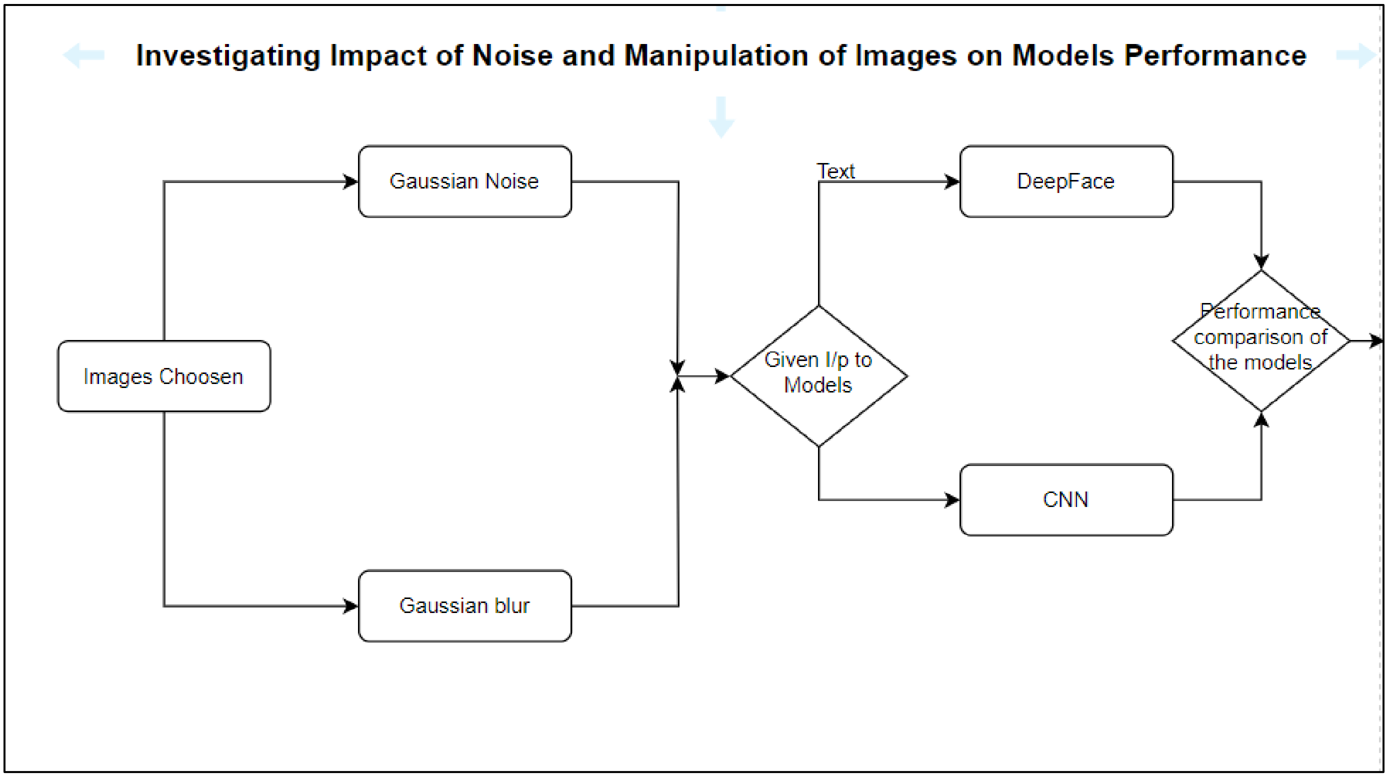
Figure [4]: Images samples from the dataset

DATAFLOW DIAGRAMS









METHODOLOGY

The methodology section describes in our project encompasses various techniques and approaches, along with discussions on inputs, outputs, and transformations:

A.) Facial Feature Extraction using Haar-cascade and OpenCV Annotations: We utilized Haar-cascade in OpenCV [13] for facial feature extraction, avoiding the need for extensive image training. This method relies on a pre-defined Haar-cascade available in an XML [14] file for detecting faces and eyes in both images and real-time webcam feeds.

B.) CNN (Convolutional Neural Network): This neural network-based approach involves data input shaped as (48,48,1). We employed ReLU activation functions in all layers except the final layer, where we used a softmax function. The CNN architecture comprises 6 convolutional layers, with 1 max pooling layer following every 2 convolutional layers, and two dense layers. We trained the neural network over various epochs (10, 15, 20, 50, 100) to determine the optimal configuration. The trained model was saved in .h5 format for further utilization in predicting emotions in unseen data and real-time recognition.

C.) DeepFace Framework: DeepFace, an open-source Python framework, was utilized for facial detection, recognition, and verification. The framework, installable via pip (pip install deepface), incorporates multiple face recognition models like VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib, and SFace. We defaulted to VGG-Face within DeepFace, which provides not only emotion analysis but also facial attribute analysis such as age, gender, and race.

D.) Real-time Emotion Recognition: For real-time emotion recognition, the DeepFace model utilized an in-built stream function. Conversely, the CNN Model utilized Python's OpenCV library [15]. We created a custom function to access the system's camera, and the CNN model was then invoked to predict emotions in real-time, accompanied by face and eye annotation.

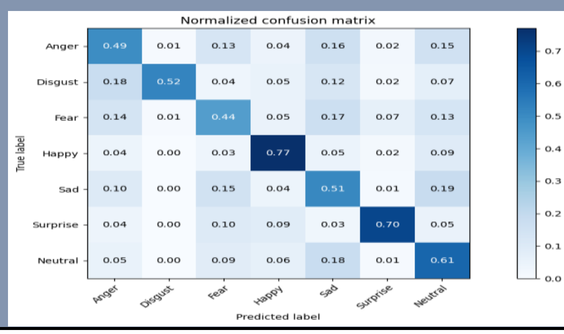
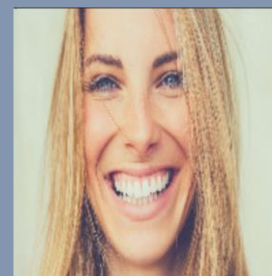
CONCLUSION

After training the models CNN running this model with Deep Face and predicting the emotions, we were able to determine the overall accuracy as illustrated below:

<table>
<tr><th>Models</th><th>Accuracy</th><th>Predictions</th></tr>
<tr><td>CNN</td><td>60.43%

from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report,
testing_true = np.argmax(n_test, axis=1)
testing_pred = np.argmax(n_pred, axis=1)
print("CNN Model Accuracy on test set: {:.4f}".form

CNN Model Accuracy on test set: 0.6043


</td><td>
happy</td></tr>
<tr><td>DeepFace</td><td>89.33%

[{'emotion': {'angry': 2.7999237931728027e-08,
'disgust': 5.2976387287820986e-11,
'fear': 99.79926942582544,
'happy': 1.3905033509643341e-08,
'sad': 4.1551280450543997e-07,
'surprise': 0.20073204100770003,
'neutral': 4.996838310639922e-10},
'dominant_emotion': 'fear',
'region': {'x': 85, 'y': 35, 'w': 88, 'h': 88},
'age': 33,
'gender': {'Woman': 0.01744396722642705, 'Man': 99.98255372047424},
'dominant_gender': 'Man',
'race': {'asian': 0.7970864903433907,
'indian': 4.68179490349232,
'black': 0.35128199492031276,
'white': 34.16132738129227,
'middle eastern': 36.95717596497249,
'latino hispanic': 23.051337665478904},
'dominant_race': 'middle eastern'}]

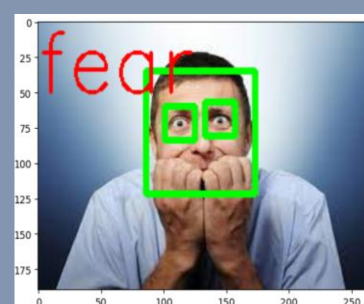
predicted[0]['dominant_emotion']</td><td>
</td></tr>
</table>

Figure [5]: Accuracies and Predictions From CNN and DeepFace

Predicting the Emotions for Group Picture:

Figure [6]: Original image

Result:
```
Number of faces detected: 5
Number of eyes detected: 10
```

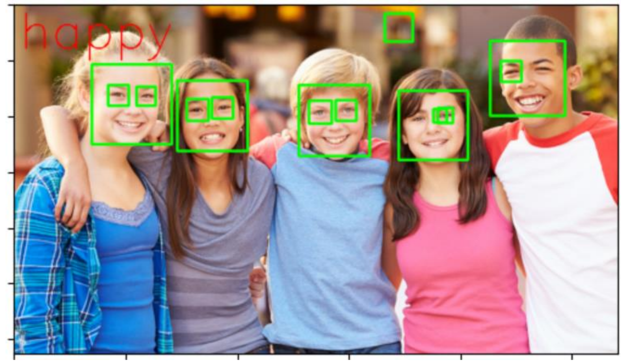
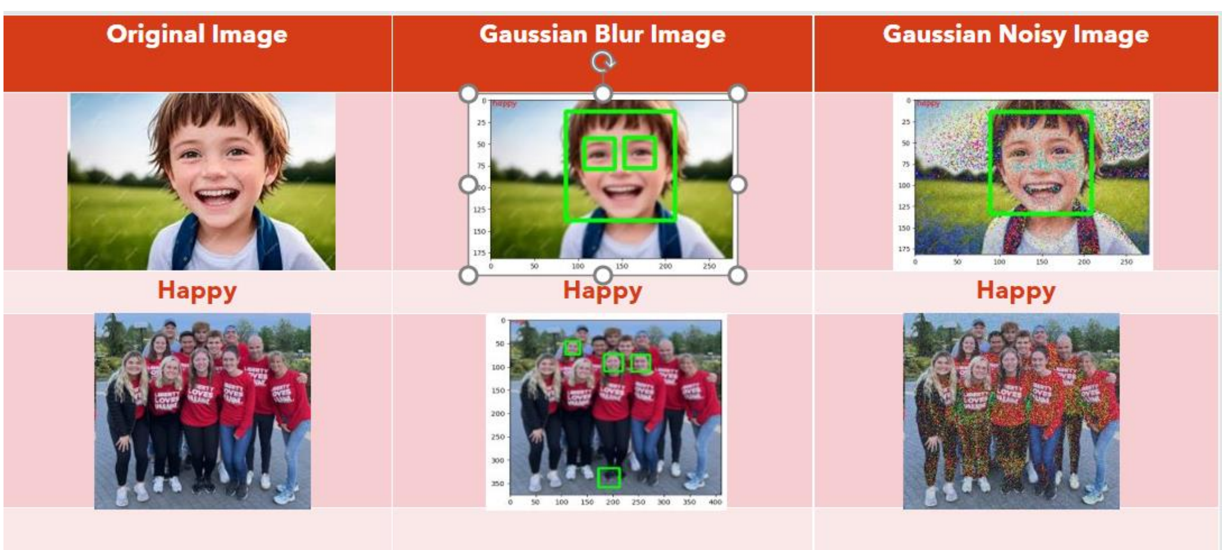


Figure [7]: Prediction from DeepFace

Happy

After achieving successful emotion predictions on images, we were keen to test our models' performance with different image qualities. We conducted experiments by manipulating image using Gaussian Blue and Gaussian Noise techniques, assessing their impact on emotion prediction accuracy in both the CNN and DeepFace models. Our findings revealed that the CNN model consistently made accurate predictions across various image alterations. However, when we incorporated gaussian blur and gaussian noise in the original image, the DeepFace model showed a tendency to generate some inaccurate predictions. A selection of these results is provided below:

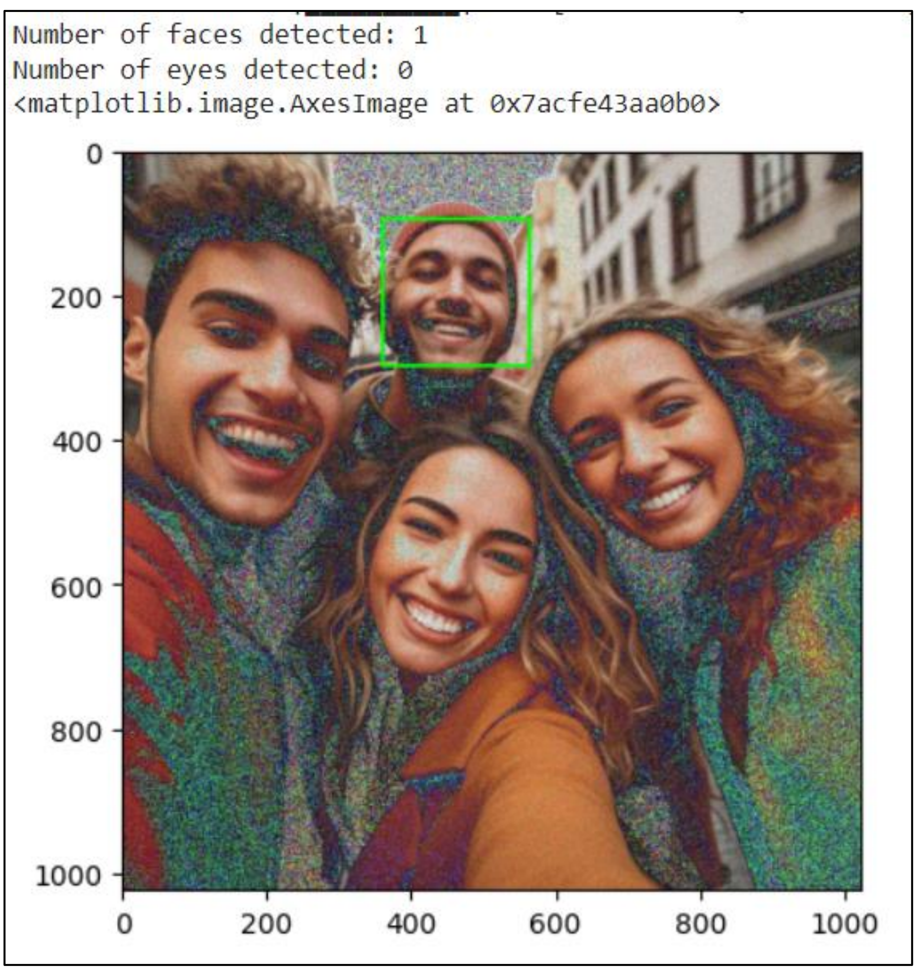
Manipulation images using DeepFace:



Manipulation Image Prediction using CNN:

CHALLENGES/LESSON LEARNED

We encountered challenges with facial annotation, particularly in group images where incorrect facial features were extracted despite successful execution on individual images. Moreover, we noticed a notable racial facial bias within our dataset. Another hurdle was the consistent lower accuracy of the CNN model, even after training it across various epochs (10, 15, 20, 50, and 100). Computational limitations also hindered our ability to further optimize the CNN model.



PROJECT PLAN (GANTT CHART)

Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
In the initial phases of our research, we conducted exploratory data analysis and implemented data preprocessing procedures to prepare and enhance the dataset for further analysis.	M1 & M2 & M3						
Distribution of emotions within each class, visualized representative images paired with their corresponding emotions, and assessed their quality as part of a comprehensive analysis.		M1	M1				
Facial recognition in each environment and extracted facial features using the Haar-cascade method within the OpenCV framework.			M2				
Implementation of Face Detection Algorithm by creating a Convolutional Neural Network model (CNN), training it, and creating a confusion matrix for the emotions.				M2			
Implementation of Face Detection Algorithm by creating a DeepFace framework-based model for prediction. Image manipulation and added Gaussian Blur and Gaussian noise images predict with DeepFace.						M1 & M2 & M3	
Prediction for still image and live/real-time feed from webcam, Documentation, Reporting and Presentation.							M1 & M2 & M3

Urvashiben Veerabhai Patel	M1
Tanmay Rewari	M2
Vinay Khandelwal	M3

REFERENCES:

[1] Ratliff M. S., Patterson E., Emotion recognition using facial expressions with active appearance models, Proceedings of the Third IASTED International Conference on Human Computer Interaction, ACTA Press, Anaheim, CA, USA, 2008, 138–143.

[2] Tian Y. I., Kanade T., Cohn J. F., Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (2001), no. 2, 97–115.

[3] Facial Recognition Technology in 2021: Masks, Bias, and the Future of Healthcare, https://link.springer.com/article/10.1007/s10916-021-01723-w.

[4] Helping disabled people: The potentials of biometric information, https://www.researchgate.net/publication/263445918_Helping_disabled_people_The_potentials_of_biometric_information.

[5] DeepFace, https://pypi.org/project/deepface/.

[6] P. Ekman, and W.V. Friesen, Facial Action Coding System (FACS): Manual. Palo Alto: Consulting Psychologists Press, 1978.

[7] Convolutional Neural Network, https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

[8] DeepFace– The Most Popular Open-Source Facial Recognition Library, https://viso.ai/computer-vision/deepface/.

[9] How to adjust brightness, contrast, sharpness and saturation of images in PIL (pillow) – with Python Examples, https://holypython.com/python-pil-tutorial/how-to-adjust-brightness-contrast-sharpness-and-saturation-of-images-in-python-pil/.

[10] Kaggle, https://www.kaggle.com/.

[11] FER2013 Dataset, https://datasets.activeloop.ai/docs/ml/datasets/fer2013-dataset.

[12] Face Detection using Haar Cascades - OpenCV-Python, https://www.numpyninja.com/post/face-detection-using-haar-cascades-opencv-python.

[13] GitHub Haar-cascade files for Image Annotation, https://github.com/opencv/opencv/tree/master/data/haarcascades.

[14] Face Detection in Python Using a Webcam, https://realpython.com/face-detection-in-python-using-a-webcam/.

[15] Facing Bias in Facial Recognition Technology, https://www.theregreview.org/2021/03/20/saturday-seminar-facing-bias-in-facial-recognition-technology/.

[16] Emotion recognition using facial expressions, https://www.researchgate.net/publication/317487657_Emotion_recognition_using_facial_expressions.

[17] Real-Time Facial Expression Recognition using Facial Landmarks and Neural Networks, https://arxiv.org/pdf/2202.00102.pdf.

[18] Emotion recognition using facial expression analysis, https://www.researchgate.net/publication/328694744_Emotion_recognition_using_facial_expression_analysis.