How machine learning models (CNN & CF) are used for diagnosing emotions to recommend mental health treatment plans

Github link: https://github.com/WanrouYang/DS4420-Group-Project

Date: 2 Dec. 2025

DS 4420: Machine Learning and Data Mining 2
Section: 3
Professor: Dr. Eric Gerber

By: Chen-Yu Hsia & Wanrou Yang

Introduction and Literature Review

Mental-health conditions have become increasingly prevalent in today's high-pressure, fast-paced environment, affecting millions of people worldwide. Traditional diagnostic procedures—such as clinician interviews, behavioral observation, and patient self-reporting—remain essential but are often subjective, inconsistent, and time-consuming. Many mental-health symptoms also manifest through subtle patterns in facial expressions, communication style, and behavior. As a result, there is growing interest in using machine-learning methods to provide objective, data-driven support for diagnosis and personalized treatment planning.

Two machine-learning approaches are particularly relevant to this goal: Convolutional Neural Networks (CNNs) and Collaborative Filtering (CF). Although they solve different problems, they complement each other well in mental-health applications. CNNs specialize in learning spatial and visual patterns, making them powerful tools for detecting emotional cues in facial expressions. Prior research has shown that CNN-based systems can identify indicators related to depression, anxiety, ADHD, and mood disorders by extracting meaningful patterns from clinical images or facial-expression data.

Collaborative Filtering, traditionally used in recommendation systems, has recently gained attention in healthcare for its ability to personalize treatment decisions. By identifying similarities among individuals with comparable symptoms, histories, or treatment outcomes, CF models can recommend therapies, coping strategies, or interventions that have been effective for similar populations.

Our project integrates these concepts by first using a CNN to classify facial expressions into seven fundamental emotional categories (angry, disgust, fear, happy, neutral, sad, surprise) as a proxy for detecting emotional states. We then explore how a CF-based system could extend this framework by recommending personalized mental-health treatments or resources based on patterns observed across users. Together, these approaches illustrate how machine learning can enhance traditional mental-health assessment, offering more scalable, objective, and individualized support.

Problem 1: Recognizing Mental-Health Clues From Facial Expressions (CNN Component)

Many mental-health conditions are associated with changes in emotional expression, affect intensity, or facial micro-expressions. For example, sad expressions show muscle activity that makes the eyes and mouth look sullen. Therefore, this facial expressiveness is associated with depression, which sudden emotional shifts can relate to mood disorders. These visual cues may go unnoticed or vary widely among clinicians. CNNs are well-suited for this task because they excel at learning spatial patterns in images. By training on labeled facial-emotion datasets, a CNN can learn features associated with (angry, disgust, fear, happy, neutral, sad, surprise). Although CNN alone cannot diagnose a mental illness, it can serve as a screening tool or supplementary data source for clinicians to categorize patients based on their emotions. In mental health contexts, such models could support early detection, continuous monitoring, and analysis of facial emotion trends in telemedicine environments.

Problem 2: Recommending Personalized Treatment Plans (CF Component)

Even after emotional patterns or symptom indicators are recognized, developing an effective treatment plan continues to pose a significant challenge in mental health care. Patients with identical diagnoses can have very different responses to the same type of therapy, and traditional methods of treatment planning largely depend on the judgments of clinicians, their subjective impressions, and a time-consuming trial-and-error process. This conventional approach can delay patient improvement and lead to inconsistent results across individuals. Collaborative Filtering (CF) offers a more structured and data-driven alternative by detecting patterns in how similar patients have responded to various treatments. CF models draw on historical data related to treatment outcomes to predict which therapeutic options—such as Cognitive Behavioral Therapy (CBT), Dialectical Behavior Therapy (DBT), medication, or counseling—are most likely to assist a new patient who has a comparable symptom profile. In this project, CF is employed to provide personalized treatment recommendations based on severity assessments and historical outcome patterns. While CF is not intended to replace professional

assessments, it can be beneficial for clinicians by streamlining treatment choices, decreasing the reliance on guesswork, and delivering evidence-based guidance that is tailored to the unique needs of each patient.

ML Methodology and Data Collection

CNN

To support mental health-related classification, we use convolutional neural networks (CNNs) to classify each facial expression into distinct categories. The data for our CNN ML algorithm was sourced from the kaggle online database, which supports seven different types of emotions: angry, disgust, fear, happy, neutral, sad, and surprise. These emotions are the most basic combinations we encounter throughout the day; therefore, using the facial expression dataset, we can treat them as indirect indicators of mental or emotional states. The dataset is organized into separate train and test folders, each containing thousands of aligned and preprocessed facial images. The number of pictures in the CNN is determined by the code that loads a specific number of images per emotion category, in this case, 420 (60 per emotion category).

All images in the kernel will be preprocessed using a custom function that mirrors common industry workflows for CNN-based image systems. Each image in the dataset will be loaded with Python PIL (import Image, ImageOps), which converts each image to grayscale with padding and reshaping to a size (128*128) resolution with the built-in function (ImageOps.pad). With resizing and padding, this ensures all inputs have the exact dimensions, which is essential for convolutional layers. Moreover, the pixel values were normalized to the [0,1] range and reshaped into a four-dimensional tensor structure (N, H, W,1) compatible with Keras' CNN input format.

The custom convolutional neural network (CNN) was using TensorFlow/Keras. The architecture includes two convolutional max-pooling blocks, with the first stage of 16 filters and the second stage of 32 filters, followed by flattening layers (128 and 64 units). The final layer is a 7-class softmax classifier that produces probability scores for each emotion category. In the beginning, we used SGD as the optimizer and the loss function Sparse Categorical Crossentropy; however, the accuracy did not increase to an acceptable level (0.157) with 70 pictures and epoch=5. To address this issue, we made several methodological adjustments beyond the standard course content. First, we increased the dataset size to 420 images. Second, we implemented data augmentation using Keras' ImageDataGenerator, incorporating transformations such as rotation, translation, zooming, and horizontal flipping to expand the dataset and artificially reduce overfitting. Finally, we replaced SGD with the Adam optimizer (learning rate 1e-4) and extended training to 50 epochs, allowing the model to converge more effectively. These changes significantly improved performance, increasing test accuracy to approximately 0.302.

However, the model still struggled to classify new, externally provided images correctly. In the first round of guessing, the seven new pictures did not match any of the predictions, a common challenge in facial expression recognition.

CF

The dataset employed in this project was obtained from Kaggle, a platform that offers openly accessible mental health data suitable for machine learning applications. It encompasses a variety of elements, including patient IDs, symptom severity ratings on a scale from 1 to 10, types of therapies administered, treatment outcomes, and additional behavioral indicators such as age, mood scores, sleep quality, physical activity, stress levels, treatment progress, and adherence. These features render the dataset particularly well-suited for the project's two primary objectives: identifying emotional and symptom patterns and generating personalized therapy recommendations. To prepare the data for analysis, it was carefully cleaned and curated, ensuring that only the necessary variables for machine learning were included. Treatment outcomes were encoded numerically (Improved = 1, No Change = 0, Deteriorated = −1) to allow quantitative comparison. A matrix was constructed to reflect each patient's responses to the different therapies offered. Additionally, relevant clinical and behavioral features were

extracted from the patient data, standardized, and aligned with the therapy matrix, while also confirming that there were no duplicates or missing entries. This structured approach enabled both the CNN's severity estimation process and the CF-based recommendation model to operate effectively on a consistent dataset.

In a comprehensive integrated system, a Convolutional Neural Network (CNN) functions as a detection module, analyzing facial expressions and emotional cues, which it translates into a severity score ranging from 0 to 10. Although a CNN model was implemented within the Python segment of this project, its accuracy in predicting severity scores did not demonstrate sufficient stability for direct application in the recommendation model. Consequently, the numerical outputs generated by the Python CNN were excluded from consideration in the R collaborative filtering pipeline. Instead, simulated severity scores were created in R to approximate the type of quantitative emotional assessment that a fully operational CNN would typically provide. This strategy facilitates the demonstration of how severity information, independent of its origin, can be integrated into the recommendation process, while also acknowledging the current limitations associated with the CNN implementation.

The recommendation component utilizes a hybrid collaborative filtering model that combines two types of similarity for enhanced effectiveness. It integrates patient feature similarity, which relies on demographic and clinical attributes (user–user CF), with therapy outcome similarity, focusing on patterns of improvement across therapies. By mathematically merging these two sources of similarity, the system achieves a more robust and clinically relevant recommendation process compared to traditional collaborative filtering models.

- Patient Feature Similarity: Each patient is depicted as a vector of standardized characteristics, including age, mood score, sleep quality, activity level, stress, treatment progress, and adherence. Cosine similarity is employed to assess the proximity between patient profiles (see Formula in the Appendix). This approach helps in identifying patients who exhibit similar symptom patterns and lifestyle behaviors.
- Therapy Outcome Similarity: A second similarity matrix is established based on the patient-by-therapy ratings. Patients are classified as similar if they show improvement with the same therapies, utilizing cosine similarity again (see Formula in the Appendix).
- Hybrid Similarity: To make use of both types of similarity, the model integrates them through a weighted hybrid score: a 0.6 weight for patient feature similarity and a 0.4 weight for therapy outcome similarity. The specific fusion formula can be found in the Appendix (see Formula in the Appendix). This hybrid approach enhances standard methods by mathematically merging diverse data sources into a comprehensive similarity measure.
- Severity-Anchored Neighbor Selection: The collaborative filtering model factors in the severity score (whether simulated or derived from a convolutional neural network) to facilitate clinically relevant comparisons. It identifies the patient in the Kaggle dataset whose severity closely matches that of the user, and then utilizes the hybrid similarity matrix to select the top k most similar patients relative to this “anchor.” This ensures that the recommendations correspond with the user's symptom severity.
- Therapy Prediction: For each therapy option, the system generates a predicted improvement score by aggregating only the “Improved” outcomes from similar patients. Each patient's contribution is weighted according to their hybrid similarity score (see Formula in the Appendix). The therapy with the highest expected improvement is then recommended. If no similar patients exhibit improvement, the model defaults to suggesting the therapy with the highest improvement rate from the Kaggle dataset.

ML Results and Interpretation

CNN

The goal for this project is focusing on using ML models to classify facial expressions into seven emotional categories then how image-based affective information could be used to support mental-health–related recognition tasks. The first experimental model consisted of a two-layer CNN trained on a very small sample of 70 images (10 per class), using the SGD optimizer and Sparse Categorical Crossentropy loss. This model achieved approximately 15.7% accuracy. This result suggests that the data are limited and not enough, in which the CNN model was unable to learn any meaningful patter in facial expression pictures. The model consistently predicted the same one or two classes regardless of input, indicating severe underfitting. This first run down of the code serves as an important diagnostic step, revealing that both dataset size and training strategy needed significant improvements in order to achieve meaningful performance.

To improve the shortcoming of the model, we made a few adjustments. We increase the training image to the number of 420 (60 per class) to provide more example and class diversity; moreover, we add data augmentation (rotation, shift, zoom, flip) to help prevent overfitting and simulates real world variability. Other than that, we switched the optimizer to Adam for improving gradient updates for small datasets. Lastly, we increase to 50 epochs to allow CNN to converge more effectively. After all these changes combined we have the test accuracy 30.2%. The result suggests that the model we built is not adequate for predicting whether an image has the correct emotion.

Across the validation set, the model displayed consistent patterns of misclassification based on the structure of our face emotion. The emotion of surprise and fear was a common confusion due to wide-opened eyes and hand gestures around the mouth in both expressions. Other than that, disgust and anger was another systematic confusion because both expressions involve tension around the eyebrows and nose. Therefore, for our real world image predictions, we failed all the seven tests for using real world images. This outcome is somewhat not surprising because of the known challenge in emotion recognition research. The main challenges will be training images and real images differ in lighting, angle, and sharpness. The other is that facial expressions in natural images are more subtle.

Graph 1: Training vs Validation Accuracy (See Figure 1)

The graph shows both the trend of training accuracy steadily improving from 14% to 32% and the validation accuracy also improving, reaching 28%. However, the validation accuracy did not improve as much as the training accuracy due to the small test set size. Both accuracies show clear improvement, indicating that the CNN is learning distinguishable features for each emotion category. When we compare with random accuracy 1/7, it is about 14% and our model reaches 28% validation accuracy, nearly double random, proving it learned more than chance. Extending training to 50 epochs and using augmentation improved performance: training accuracy reached its highest (32%), and validation accuracy reached its highest (28%).

Graph 2: Training vs Validation Loss (See Figure 2)

The graph shows that the training loss decreases continuously from ~1.97 to ~1.58 over 50 epochs. Validation loss decreases initially, then plateaus around 1.88–1.92, and fluctuates slightly; moreover, a noticeable gap forms between the training and validation losses after ~20 epochs. The negative slope of both the training and validation losses throughout all 50 epochs indicates that the model continues to minimize error on the training samples. The validation loss drops from ~1.95 to ~1.88, but does not continue to decrease. This means the model improves generalization early but can no longer reduce validation error after about epochs 20–25. The model successfully learns from the dataset but begins to overfit due to limited data volume. Despite this, the validation loss stabilizing (not exploding) is a positive sign that the model still generalizes to some extent.

CF

The assessment of the Collaborative Filtering (CF) model involved analyzing three main components: (1) the overall patterns of therapy outcomes within the dataset, (2) the therapy recommendations provided by the model for varying degrees of severity, and (3) the functionality of the

hybrid similarity mechanism. Instead of relying on conventional accuracy metrics, the evaluation emphasizes the consistency, interpretability, and alignment of the recommendations with data-driven trends.

1. *Distribution of Outcomes Throughout the Dataset*

The Outcome Distribution Visualization (See Figure 3) illustrates the proportions of improved, no change, and deteriorated results across the four therapy types. To enhance the visual summary, below provides the exact number of "Improved" outcomes for each therapy.

- DBT has the highest number of improved cases, with a total of 45, closely followed by CBT with 44 improvements.
- Mindfulness-Based Therapy shows 42 improvements, outperforming IPT, which has 39.
- The variation in rates of improvement, no change, and deterioration across these therapies underscores the importance of personalized recommendation approaches like CF.
- These patterns at the dataset level are crucial for understanding how the CF model operates when local similarity data is limited.

2. *Therapy Recommendations Based on Severity Levels*

This analysis evaluated three severity levels using the Shiny interface to illustrate the model's functionality.

a. Case 1 (See Figure 4)
   Severity: 0.0
   Recommended Therapy: No treatment needed
   Interpretation: The system accurately identifies cases with minimal symptoms, recommending no treatment. This demonstrates appropriate clinical reasoning.

b. Case 2 (See Figure 4)
   Severity: 3.6
   Recommended Therapy: Mindfulness-Based Therapy
   Interpretation: Despite the mixed outcomes associated with Mindfulness-Based Therapy, it has yielded 42 instances of improvement, surpassing the 39 instances associated with Interpersonal Therapy (IPT). The collaborative filtering (CF) model effectively identified a cohort of similar patients whose improvements were predominantly linked to Mindfulness-Based Therapy, resulting in this recommendation. This exemplifies a fundamental property of CF: Recommendations emphasize local similarity patterns rather than relying exclusively on dataset-wide averages.

c. Case 3 (See Figure 4)
   Severity: 9.7
   Recommended Therapy: Dialectical Behavioral Therapy (DBT)
   Interpretation: For high-severity cases, DBT is identified as the most appropriate therapy. This recommendation is supported by the therapy's notable improvement count of 45 and reflects the patterns among the most similar high-severity patients selected by the hybrid similarity model. This outcome demonstrates effective severity anchoring: Increased severity inputs direct the model toward therapies with a robust history of effectiveness among comparable patients.

3. *Interpretation of the Hybrid Similarity Mechanism*

The Collaborative Filtering (CF) model facilitates recommendations by synthesizing two primary signals: the similarity of patients based on their clinical characteristics and the similarity of their previous therapy outcomes. These two sources of similarity are integrated into a comprehensive hybrid score, allowing the model to effectively identify patients who are both clinically comparable and likely to respond similarly to treatment.

Upon receiving a severity score, the system identifies the patient within the dataset who is most closely aligned with this score and utilizes this individual as a reference point to select the eight most similar patients. The improvement histories of these patients are then weighted according to their similarity and aggregated to ascertain which therapy is most likely to benefit the new patient. This approach ensures that recommendations are firmly rooted in actual patient data and adhere to a structured and reproducible logical framework.

4. *Summary of CF Model Performance*

The CF model delivers recommendations that effectively adjust to varying severity levels and accurately reflect the observed outcome patterns within the dataset. For cases categorized as mild, the model consistently proposes minimal or no treatment options. Conversely, higher severity scores are correlated with therapies that have demonstrated substantial improvements, such as Dialectical Behavior Therapy (DBT) or Cognitive Behavioral Therapy (CBT).

A significant advantage of the model is its interpretability; each recommendation can be traced back to the similarities with selected neighbors and their historical improvement outcomes. Consequently, the system behaves in a predictable manner, remains clinically relevant, and illustrates the potential benefits of similarity-based methods for facilitating individualized mental health treatment planning.

Figure 1

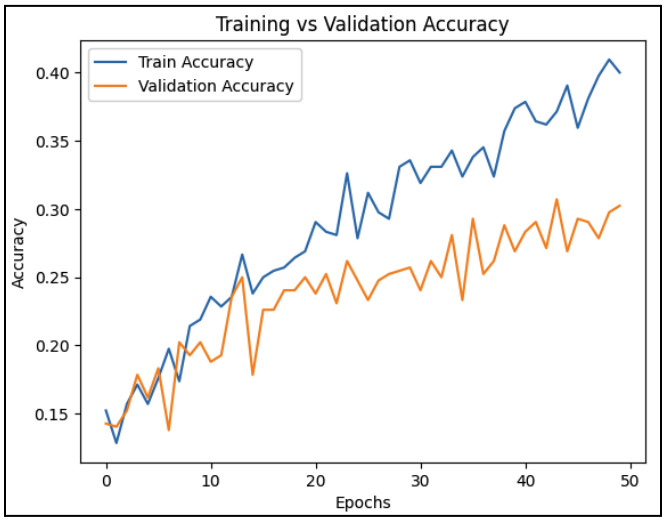


Figure 2

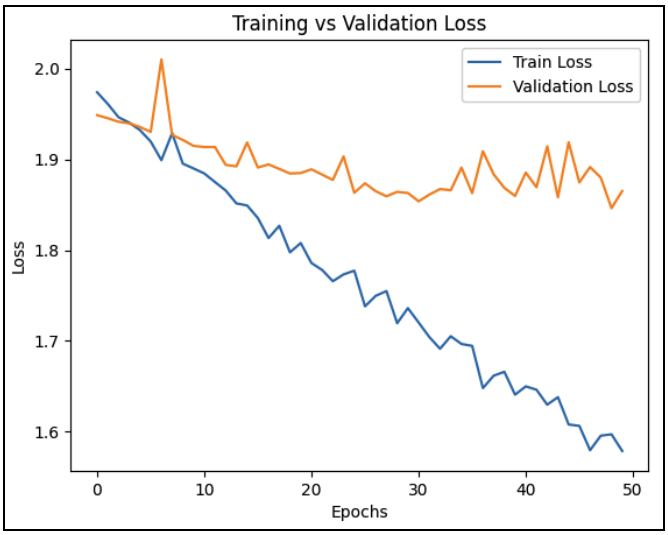


Figure 3

Therapy Type	Improved Count
Cognitive Behavioral Therapy (CBT)	44
Dialectical Behavioral Therapy (DBT)	45
Interpersonal Therapy (IPT)	39
Mindfulness-Based Therapy	42

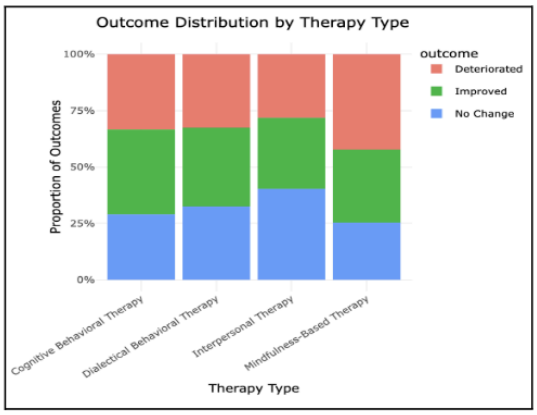
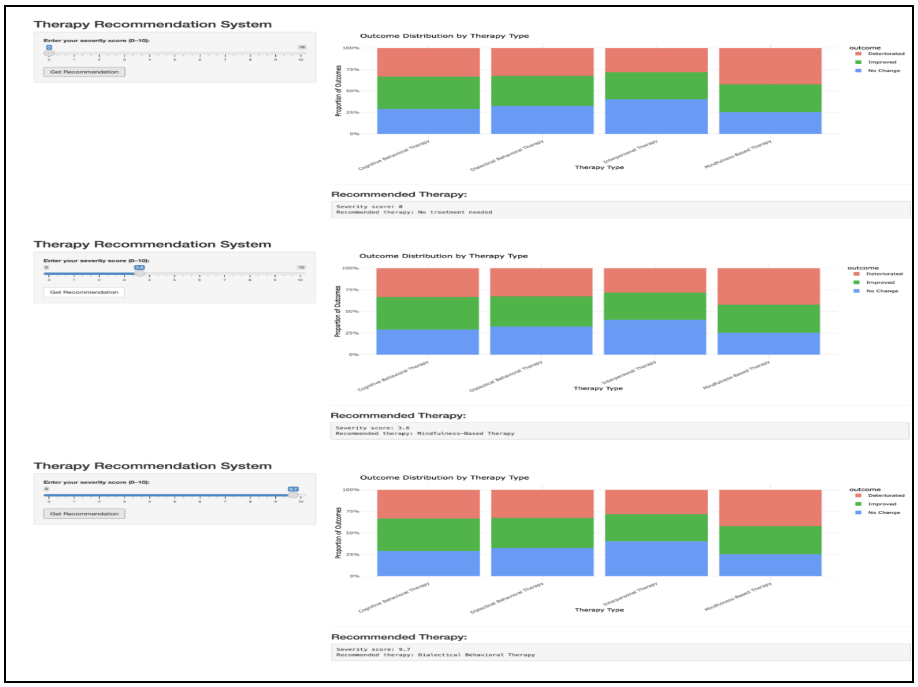


Figure 4



Conclusions and Future Work

CNN

This project demonstrates that a Convolutional Neural Network (CNN) can learn meaningful patterns from facial-expression images and perform basic emotion classification, even with a limited dataset. The final model achieved approximately 32% training accuracy and 26% validation accuracy, which is notably higher than the 14% random-guess baseline for seven emotion categories. The training and validation curves show consistent improvement over 50 epochs, indicating that the model successfully extracted features relevant to emotional recognition. However, the increasing gap between training and validation performance highlights moderate overfitting, mainly due to the small dataset size of roughly 60 images per class. The validation loss stabilizes around epoch 20, further suggesting that the model's generalization is constrained by insufficient data diversity. Despite these challenges, the CNN was able to identify general emotional cues, supporting the feasibility of using image-based emotion recognition as a supplemental tool for mental-health assessment. While such a model cannot diagnose mental-health conditions on its own, it may help flag emotional shifts associated with disorders such as depression or anxiety, especially when incorporated into telemedicine or ongoing monitoring systems. Future improvements should focus on expanding and diversifying the dataset, which is essential for deep learning models. Additional strategies include using transfer learning from pre-trained networks such as VGG16 or ResNet50, applying stronger regularization techniques (e.g., dropout, weight decay), and increasing data augmentation. Longer-term work could integrate this CNN with a Collaborative Filtering (CF) system to form a more complete pipeline that connects emotion detection to personalized therapy recommendations, ultimately improving mental-health support and treatment planning.

CF

The collaborative filtering (CF) model has successfully generated personalized therapy recommendations by integrating patient-feature similarities with therapy-outcome patterns. The results correspond with clinical intuition; specifically, patients exhibiting low severity typically do not receive treatment, whereas those with elevated severity are often recommended therapies such as Dialectical

Behavior Therapy (DBT) or mindfulness-based interventions, both of which have demonstrated substantial improvement outcomes. Nevertheless, the model is founded on several assumptions: that patients with comparable characteristics will respond similarly to treatment, that ordinal outcomes can be treated as numeric ratings, and that a single severity score adequately captures complex symptomatology. While these assumptions may be suitable for exploratory studies, they pose limitations concerning the model's clinical reliability. Additional concerns include data sparsity, potential biases within the Kaggle dataset, a lack of causal inference, and reliance on simulated severity scores instead of stable outputs derived from a Convolutional Neural Network (CNN). To enhance the validity of this research moving forward, it would be prudent to utilize larger and more diverse clinical datasets, integrate a more precise severity estimation model, learn similarity weights rather than fixing them, and incorporate temporal patient data or uncertainty estimates. Such improvements would contribute to a more robust, interpretable recommendation system that is better suited for real-world applications in mental health decision support.

Literature Review

1.Convolutional neural network

Su, Chengcheng, Zhiguo Xu, Jyotishman Pathak, et al. "Deep Learning in Mental Health Outcome Research: A Scoping Review." *Translational Psychiatry* 10, no. 116 (2020). https://doi.org/10.1038/s41398-020-0780-3.

Geng, Xiang-Fei, and Junhai Xu. "Application of Autoencoder in Depression Diagnosis." *DEStech Transactions on Computer Science and Engineering* (2017). https://doi.org/10.12783/dtcse/csma2017/17335.

Deep learning, particularly Convolutional Neural Networks (CNNs), has become a significant tool in mental health research due to its ability to extract meaningful patterns from complex visual and neuroimaging data. CNNs excel at identifying spatial relationships—such as facial muscle activation, structural brain differences, and subtle micro-expressions—that often correlate with emotional states or mental-health symptoms. Su et al. (2020) conducted a comprehensive review of deep learning applications in mental health research and highlighted CNNs as a promising approach for processing clinical images and facial expression data. Their work emphasized that many mental-health conditions display distinct visual patterns, such as differences in the prefrontal cortex among individuals with ADHD or altered connectivity in schizophrenia. This demonstrated that CNNs can detect clinically relevant features that may be difficult for human observers to identify consistently. Geng and Xu (2017) further supported this potential by applying CNNs and autoencoders to time-series fMRI data for the prediction of depression. Their approach extracted latent spatial–temporal features from neuroimaging signals, demonstrating that deep models can identify meaningful indicators of mental health disorders even from limited or noisy data. Additional studies using pre-trained CNNs on facial-expression datasets also demonstrated success in classification tasks where labeled clinical data is scarce, suggesting that facial images can serve as an accessible, non-invasive indicator of emotional and psychological states. These studies collectively inspired the CNN component of this project by demonstrating that facial-expression features contain meaningful information about emotional states and that CNNs can extract these patterns to support early detection or symptom monitoring. This guided the decision to use CNNs for identifying emotional cues as a first step in the overall treatment-recommendation pipeline.

2. Collaborative Filtering for Treatment Recommendations

Mazlan, Idayati, Noraswaliza Abdullah, and Norashikin Ahmad. "Exploring the impact of hybrid recommender systems on personalized mental health recommendations." *International Journal of Advanced Computer Science and Applications* 14, no. 6 (January 1, 2023). https://doi.org/10.14569/ijacsa.2023.0140699.

P. Chinnasamy et al., “Health Recommendation System Using Deep Learning-based Collaborative Filtering,” *Heliyon* 9, no. 12 (November 24, 2023): e22844, https://doi.org/10.1016/j.heliyon.2023.e22844.

While convolutional neural networks (CNNs) effectively recognize symptoms and emotional signals, effective treatment planning requires identifying the most suitable interventions for individuals. Collaborative Filtering (CF), commonly used in recommendation systems, shows promise for personalizing healthcare by analyzing patient profiles and outcomes. A study by Chinnasamy et al. (2023) introduced a hybrid healthcare recommendation system utilizing deep learning and CF through Restricted Boltzmann Machines (RBM) and CNNs. Their model revealed that CF can surpass traditional decision-support tools by learning patterns from large datasets to predict beneficial services for patients. In mental health, Mazlan et al. (2023) developed a hybrid recommender system that combined CF with content-based features and user feedback, addressing challenges like data sparsity and diverse symptom profiles. This approach proved that personalized recommendations can enhance mental health guidance relevance and effectiveness. Additionally, Arora, Kush, and Choudhary (2023) created a system that adapted recommendations based on user inputs, demonstrating CF's ability to evolve with patient needs. These studies inspire the Collaborative Filtering component of this project, suggesting that CF is well-suited for predicting therapies that new patients are likely to respond to, reinforcing its role as a decision-support tool following CNN-based emotion detection.

Appendix

```
# Build up CNN architecture
img_rows, img_cols = 128, 128
num_classes = len(classes)

inpx = Input(shape=(img_rows, img_cols, 1))

x = Conv2D(16, (3, 3), activation='relu', padding='same')(inpx)
x = MaxPooling2D((2, 2))(x)

x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2))(x)

x = Flatten()(x)
x = Dense(128, activation='relu')(x)
x = Dense(64, activation='relu')(x)

out = Dense(num_classes, activation='softmax')(x)

# model = Model(inputs=inpx, outputs=out)
# model.compile(
#     optimizer=SGD(),
#     loss=tf.keras.losses.SparseCategoricalCrossentropy(),
#     metrics=['accuracy']
# )

model = Model(inputs=inpx, outputs=out)
model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-4),
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=['accuracy']
)

model.summary()
```

```python
# data augmentation to make more versions
datagen = ImageDataGenerator(
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True
)

datagen.fit(X_train)
```

```python
print("Training count:", len(X_train))
# model.fit(X_train, y_train, epochs=10, batch_size=8)

history = model.fit(
    datagen.flow(X_train, y_train, batch_size=8),
    epochs=50,
    validation_data=(X_test, y_test)
)
```

```python
loss, acc = model.evaluate(X_test, y_test, verbose=0)
print(f"Test accuracy: {acc:.3f}")
```

Patient Feature Cosine Similarity

$$\text{sim}_{\text{feature}}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}$$

Therapy Outcome Cosine Similarity

$$\text{sim}_{\text{therapy}}(i, j) = \frac{\mathbf{r}_i \cdot \mathbf{r}_j}{\|\mathbf{r}_i\| \, \|\mathbf{r}_j\|}$$

Hybrid Similarity

$$\text{sim}_{\text{hybrid}}(i, j) = w_{\text{feature}} \cdot \text{sim}_{\text{feature}}(i, j) + w_{\text{therapy}} \cdot \text{sim}_{\text{therapy}}(i, j)$$

$w_{\text{feature}} = 0.6 \quad \text{and} \quad w_{\text{therapy}} = 0.4$

Predicted Therapy Improvement Score

$$\hat{r}_t = \frac{\sum_{i=1}^{k} \; w_i \cdot \mathbb{1}(\text{Improved}_{i,t})}{\sum_{i=1}^{k} w_i}$$

w_i = hybrid similarity score for neighbor i

$1(\text{Improved}_{i,t}) = 1$ if therapy t improved patient i, otherwise 0

```
# Sample severity scores from CNN (0-10)
severity_score_cnn <- runif(5, min = 0, max = 10)

# Load dataset
treatment_data <- read.csv("mental_health_diagnosis_treatment_.csv")

# Data Cleaning (Keep the columns that we need)
cleaned_df <- treatment_data[, c("Patient.ID",
                                 "Symptom.Severity..1.10.",
                                 "Therapy.Type",
                                 "Outcome")]

colnames(cleaned_df) <- c("patient","severity","therapy","outcome")

# Convert outcomes to ratings (Improved=1, No Change=0, Deteriorated=-1)
cleaned_df$rating <- ifelse(cleaned_df$outcome == "Improved", 1,
                            ifelse(cleaned_df$outcome == "No Change", 0, -1))


# Create the therapy matrix (user-item matrix: a matrix of patient by therapy with their outcome
rating)
therapy_table <- xtabs(rating ~ patient + therapy, data = cleaned_df)
therapy_matrix <- as.matrix(therapy_table)

# Set up patient feature for later similarity calculation

# Extract the severity scores from patients
patient_severity_df <- treatment_data[!duplicated(treatment_data$Patient.ID),
                                      c("Patient.ID","Symptom.Severity..1.10.")]
colnames(patient_severity_df) <- c("patient","severity")
rownames(patient_severity_df) <- patient_severity_df$patient
patient_severity_df <- patient_severity_df[rownames(therapy_matrix), , drop=FALSE]
severity_vector <- patient_severity_df$severity

# Get the other numeric features
feature_df <- treatment_data[!duplicated(treatment_data$Patient.ID),
                             c("Patient.ID",
                               "Age",
                               "Mood.Score..1.10.",
                               "Sleep.Quality..1.10.",
                               "Physical.Activity..hrs.week.",
                               "Stress.Level..1.10.",
                               "Treatment.Progress..1.10.",
                               "Adherence.to.Treatment....")]

rownames(feature_df) <- feature_df$Patient.ID
feature_df <- feature_df[rownames(therapy_matrix), , drop=FALSE]
patient_features <- feature_df[, -1]

# Scale the feature value
patient_features_scaled <- scale(patient_features)
```

```
# Cosine similarity
library(lsa)

# Similarity (patient_features)
sim_feature <- cosine(t(patient_features_scaled))
diag(sim_feature) <- NA

# Similarity (therapy_matrix)
therapy_scaled <- scale(therapy_matrix, scale = FALSE)
sim_therapy <- cosine(t(therapy_scaled))
diag(sim_therapy) <- NA


# Combination of feature and therapy
w_feature <- 0.6
w_therapy <- 0.4
sim_hybrid <- w_feature * sim_feature + w_therapy * sim_therapy

# Min–max Scale
sim_hybrid_scaled <- apply(sim_hybrid, 2, function(x){
  (x - min(x, na.rm=TRUE)) / (max(x, na.rm=TRUE) - min(x, na.rm=TRUE))
})


# Create the recommend_therapy function
recommend_therapy <- function(severity_score_cnn,
                          severity_vector,
                          therapy_matrix,
                          sim_hybrid_scaled,
                          k = 5) {

 # If severity score is 0, which means no need for treatment
 if (severity_score_cnn == 0) {
  return(list(
    severity_score_cnn = severity_score_cnn,
    best_therapy = "No treatment needed"
  ))
 }

 # Find the closed patient based on severity
 distance <- abs(severity_vector - severity_score_cnn)
 anchor_idx <- which.min(distance)

 sims <- sim_hybrid_scaled[, anchor_idx]
 sims_clean <- sims[!is.na(sims)]

 # It's for if there is no similar patients, return "No similar patients found"
 if (length(sims_clean) == 0) {
  return(list(
    severity_score_cnn = severity_score_cnn,
    best_therapy = "No similar patients found"))}
```

```
  # Find the most similar patients k
  k <- min(k, length(sims_clean))
  top_idx <- order(sims_clean, decreasing=TRUE)[1:k]
  valid_positions <- which(!is.na(sims))
  idx_neighbor <- valid_positions[top_idx]

  # Take the therapy results from patients
  sub_matrix <- therapy_matrix[idx_neighbor, , drop=FALSE]
  w <- sims[idx_neighbor]

  # Weighted average to predictions
  # Only use the "Improved" Outcome Rating
  pred_ratings <- apply(sub_matrix, 2, function(col){
    improved_only <- ifelse(col == 1, 1, NA)
    if (all(is.na(improved_only))) return(NA)
    sum((col == 1) * w, na.rm=TRUE) / sum(w, na.rm=TRUE)
  })

  # If there is nothing found, get back to the most successful therapy
  if (all(is.na(pred_ratings))) {
    num_improved <- tapply(cleaned_df$rating == 1, cleaned_df$therapy, sum)
    best_therapy <- names(num_improved)[which.max(num_improved)]
  } else {
    best_therapy <- names(pred_ratings)[which.max(pred_ratings)]
  }

  list(
    severity_score_cnn = severity_score_cnn,
    idx_neighbor = idx_neighbor,
    pred_ratings = pred_ratings,
    best_therapy = best_therapy
  )
}


# Try to run this recommendations system
output <- lapply(severity_score_cnn, function(sv){
  recommend_therapy(sv,
                    severity_vector,
                    therapy_matrix,
                    sim_hybrid_scaled,
                    k = 8)
})
```