ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

EARLY DETECTION DEPRESSION BASED ON ACTION UNIT AND EYE GAZE FEATURES USING A MULTI-INPUT CNN-WoPL FRAMEWORK

Sugiyanto Sugiyanto[1,2] , I Ketut Eddy Purnama[1,4], Eko Mulyanto Yuniarno[1,4], Mauridhi Hery Purnomo[1,3,4*]

[1]Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Indonesia
[2]Distance Learing Study Program in Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
[3]University Center of Excellence on Artificial Intelligence for Healthcare and Society, Institut Teknologi Sepuluh Nopember, Indonesia
[4]Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Indonesia

email: 07111960010006@student.its.ac.id[1], ketut@te.its.ac.id[2], ekomulyanto@ee.its.ac.id[3], hery@ee.its.ac.id[4]

Abstract

Depression is a common mental disorder with significant life impact, including a high risk of suicide. Patients with depression attempt suicide five times more often than the general population. Self-reporting, subjective judgement and clinician expertise influence conventional diagnostic methods. For timely intervention and effective treatment, early and accurate diagnosis of depression is essential. This study proposes a framework called Multi-Input CNN-WoPL, a CNN-based method without a pooling layer that combines two features - action units and gaze - to improve accuracy and robustness in automatic depression detection. Pooling layer reduces spatial dimension of feature map, resulting in loss of information related to expression data, affecting depression detection result. The performance of the proposed method results in an accuracy of 0.994 and F1 score = 0.993, the F1 score value close to 1.0 indicates that the proposed method has good precision, recall and performance.

Keywords : Depression, Multi-Input, CNN, action unit, eye gaze

Received: 31-08-202 | **Revised:** 23-10-2024 | **Accepted:** 24-10-2024
DOI: https://doi.org/10.23887/janapati.v13i3.84674

INTRODUCTION

Depression is a common mental disorder that affects millions of individuals around the world and has a profound effect on their lives: personal well-being, social relationships and economic productivity [1]. Individuals with depressive disorders are particularly at high risk of suicide [2] and suicidal behaviour [3]. Lifetime prevalence of attempted suicide in patients with MDD is approximately 31% [4], and prevalence of thoughts of suicide is nearly 38% [5]. It has been found that the incidence of suicide attempts in patients with depression is five times higher than in the general population [6].

In view of these serious risks, an early and accurate diagnosis of depression is essential for timely intervention and effective treatment. Studies have consistently shown that early intervention in depression results in improved quality of life [7,8]. Early intervention has been shown to improve overall quality of life and reduce mortality by reducing symptoms of depression [9]. Conventional diagnostic methods often rely on subjective assessment and self-reporting, which is affected by a number of factors, including disclosure and clinician expertise [10]. This is challenging for physicians when diagnostic criteria are not completely fulfilled, and the diagnosis may change as more information becomes available [11]

These are objective, nonverbal cues in facial expression and eye movement that are automatically detected and analysed without

clinician intervention. [12]. Action units are fundamental components of facial expressions. They are defined by the Facial Action Coding System (FACS). They provide valuable information about an individual's emotional state by representing the contraction or relaxation of one or more facial muscles.Action Unit patterns are associated with depressive symptoms, such as reduced facial expressivity [13] and slower head movements [14,15]. FACS defines AU intensity on a 5-point ordinal scale from A to E level [16]. Intensity has excellent predictive power for depression via facial expressions [17], and contains information about the subject's mental state, emotional involvement [18], mental complexity [19], and spontaneous facial expressions [20]. As the intensity of specific facial muscle movements is associated with different emotions, it is potentially a feature for depression detection. AU14 is the difference between objects that are depressed and objects that are not depressed [17]. Akbar et al. used 20 AUs as input to an optimised standard feed-forward neural network and achieved 97.83% accuracy [21]. Jiang et al. used the SVM method based on 17 AU features and achieved 55.8% accuracy [22]. Yu et al. used a mixed Gaussian model with 17 AUs and achieved 76.7% accuracy [23].

Eye movements can be used to differentiate depression from other mental illnesses. Eye movement is a reflection of cognitive deficits that can be helpful in the diagnosis of depression, and it can make the whole diagnostic process more accurate [24]. Eye gaze and eye movement are closely related because eye movements are the physical actions that result in the direction of the gaze. Eye gaze is output as 4 vectors, the first two describe the direction of the eyes in world coordinate space, the second two describe the direction of the eyes in head coordinate space. Shu et al. identified an association between behavioural characteristics of eye gaze and levels of depression, achieving an accuracy of 77.1% and an F1 score of 76.9 [25]. Kobo et al, using eye gaze pattern features, were able to achieve above chance classification (60+%) of depressive tendency levels [26].

Several studies have been conducted on the use of two features to detect depression. Cohn et al used facial action and vocal prosody to detect depression with 79% accuracy [27]. Wang et al. analysed the features of facial expression and eye movement for the detection of depression, and the results showed that these features could achieve a detection accuracy of 78.85% and a recall of 80.77% [28]. Ghadiri et al. combined voice and text features to detect depression and achieved an F1 score of 82.4% [29]. Li et al. used image and text features with distribution normalisation to detect depression, achieving an accuracy of 92.84% and an F1 score of 92.78% [30]. A primitive combination of human behaviours, such as the AU and the direction of gaze, can be used effectively to

Table 1. Gaps in previous research

Research	Research Gaps	Proposed Framework
Cohn [27]	• The diversity and variety of facial and vocal data is limited, leading to potential bias in model performance. • Dynamic changes in emotion, facial expression and speech in relation to depression have not been recorded. • Lack of integration with deep learning	• DAIC-WoZ dataset that provides dynamic facial expression data. • Using multimodal features: AU and eye gaze. • Ground truth from PHQ-8 questionnaire which is reliable for screening depression diagnosis. • Using simple CNN-based deep learning without layer pooling. • Automatic feature extraction
Wang [28]	• Lacks a multimodal approach. • Does not analyse in depth the dynamic changes in expression • Reliance on manual feature extraction	
Ghadiri [29]	• The complexity of graph-based features is difficult to interpret. • Difficult to generalise across datasets or languages. • Does not fully explore long-term emotional changes, which are important for detecting depression.	
Li [30]	• The model is complex, requiring significant computational resources, making it difficult to scale. • Its performance may be limited to social media platforms • Model complexity may lead to overfitting, if trained on small or biased data sets.	
Song [31]	• Feature integration is complicated, making real-time implementation more difficult. • Dataset Limitations: Limited or homogeneous datasets. • Feature combination may lead to overfitting, especially on small or specific datasets.	

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

detect depression [31]. Gaps in the results of several previous studies, as shown in Table 1.

Deep learning's ability to work with large, powerful and efficient databases will rapidly advance healthcare, transforming medicine and improving both doctor and patient experiences. [32]. Convolutional Neural Networks (CNNs) is one of the deep learning methods that has been widely used for depression classification, either monomodal (text [33], image [34], signal [35], or speech [36] or multimodal ([37], image and speech [38], or text, audio, and video [39]).

The present research proposes a framework named Multi-Input CNN-WoPL, CNN-based method without a pool layer that combines these two characteristics-action units and gaze pattern-to improve accuracy and robustness in automatic depression detection. The pooling layer reduces the spatial dimension of the feature map, resulting in the loss of significant information [40]. The loss of significant information associated with expression data will affect the results of depression detection. The aim of the research is to contribute to an efficient automatic detection of depression and thus better reduce the risk of suicide in patients with depressive disorders.

METHOD

Proposed Framework

In this proposed framework, several steps must be performed, including: Data Preparation (preparation of data in a suitable format to be used as input for the method to be used), Multi-Input CNN-WoPL (CNN-based method without pooling layer consisting of 2 models each using a single feature input and then combining the

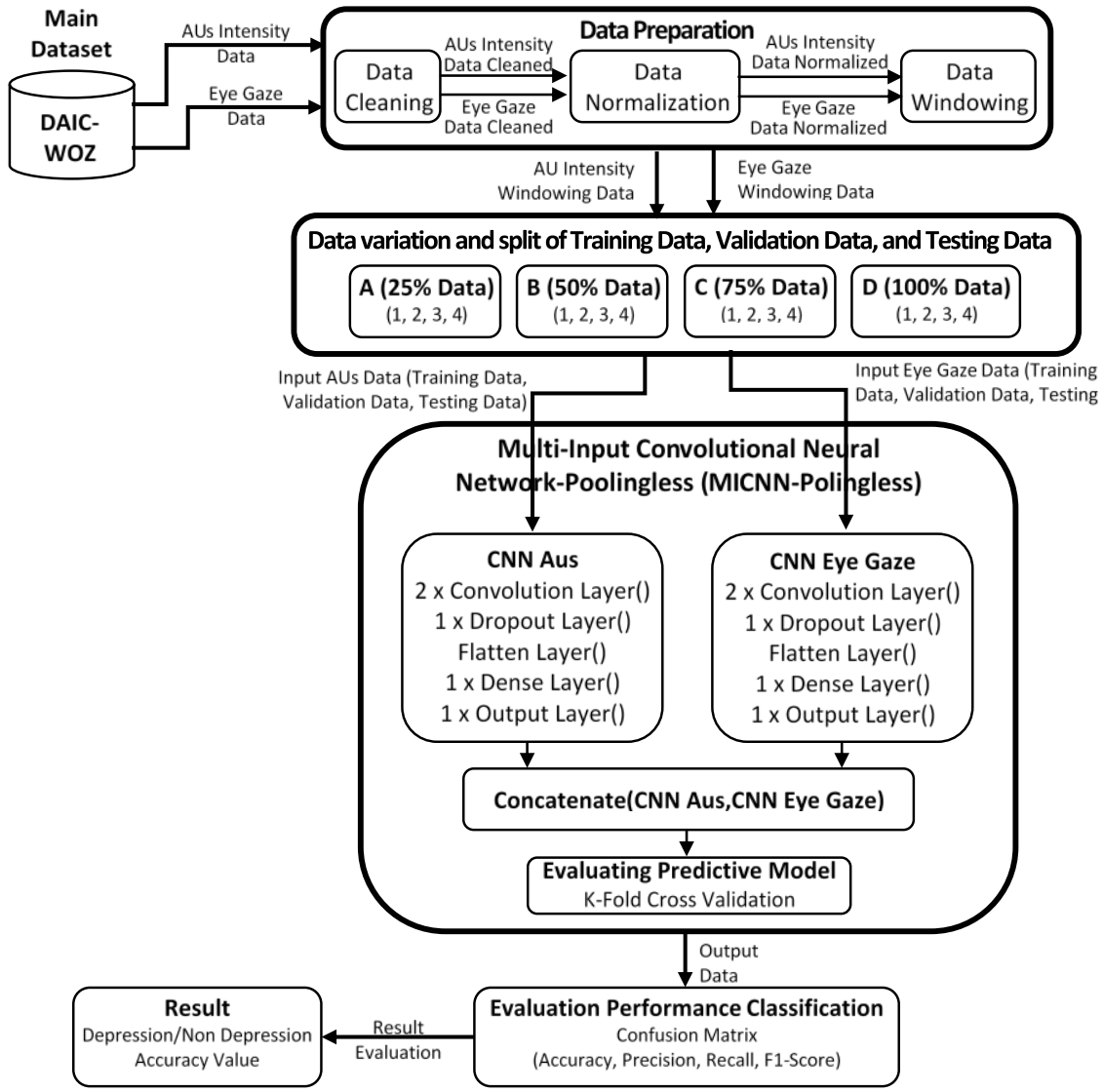

Figure 1. Multi-Input CNN-WoPL (Convolutional Neural Networks-Without Pooling Layer) Framework

outputs of both models to obtain 1 output), Model Prediction Evaluation (evaluation of model predictions during the data training process using KFold Cross Validation) and Classification Performance Evaluation (classification performance is evaluated using several values that show how well the classification results are achieved using Confusion Matrix, Accuracy Score, F1 Score and AUC Score). Detail of proposed framework, as shown in Fig. 1.

Dataset

The Distress Analysis Interview Corpus-Wizard of Oz (DAIC-WOz) dataset provides a set of features that were extracted from clinical interviews conducted with 189 participants using the OpenFace 2.0 application [41]. Labelling in this dataset uses the Patient Health Questionnaire-8 (PHQ-8), which has been established as a valid diagnostic and severity measure for depressive disorders [42]. The binary PHQ-8 is used to determine whether participants are depressed or not. In this study, the Action Unit (AU) and Eye Gaze features were used.

The AU feature in this dataset stores intensity values in a five-point ordinal scale, ranging from 0.0-5.0. The corresponding AU intensity can provide more information about a person's mental state and emotional involvement. Because people express emotions differently in different situations, AU intensity information can be used to tailor emotion recognition to specific users and situations. The AU features used were 14 AUs, as shown in Table 2.

Eye gaze is output as 4 vectors, The first two describe the direction of the eyes in world coordinate space, the second two describe the direction of the eyes in head coordinate space (i.e. if you roll your eyes, they'll point up in head coordinate space), as shown in Table 3.

Data Preparation

The proposed framework uses AU intensity and eye gaze as features data from DAIC-Woz Dataset. To prepare the data used as input, several things were done:

Table 2. Index and AU Names in DAIC-Woz Dataset

AU	Name	AU	Name
AU01	Inner Brow Raiser	AU12	Lip Corner Puller
AU02	Outer Brow Raiser	AU14	Dimpler
AU04	Brow Lowerer	AU15	Lip Corner Depressor
AU05	Upper Lid Raiser	AU17	Chin Raiser
AU06	Cheek Lid Raiser	AU20	Lip Stretcher
AU09	Nose Wrinkler	AU25	Lips Part
AU10	Upper Lip Raiser	AU26	Jaw Drop

Table 3. Eye gaze stored values in DAIC-Woz Dataset

Vectors	Description
x_0, y_0, z_0, x_1, y_1, z_1	World coordinate space vectors of both eyes.
x_h0, y_h0, z_h0, x_h1, y_h1, z_h1	Head coordinate space vectors of both eyes.

a. Data Filtering

The AU intensity data value stored in the DAIC-WOz dataset was expected to range between 0.0-5.0. However, some parts of the data have values outside the value range that will interfere with the data training process and will definitely have an effect on the results of data training and classification. Therefore, data filtering is done by eliminating data with values outside the 0.0-0.5 range. The process of eliminating inappropriate data is performed according to Algorithm 1.

The algorithm initializes an empty dataset `D'` to store the filtered data. For each participant `p` in the original dataset `D`, a set `valid_frames` is initialized to store the valid frames for that participant. For each frame `f` in the participant `p`, the algorithm checks if all feature values in `f` are within the range [0.0, 5.0] using a boolean flag

Algorithm 1: Data Filtering AU
Input: A dataset D with n participants, m features, and a variable number of frames f for each participant, where each feature value is in the range [0.0, 5.0].
Output: A filtered dataset D' containing only valid data (feature values within the range [0.0, 5.0]).

```
1. ALGORITHM DataFilteringAU(D)
2.    D'← Ø
3.    for each participant p in D
4.       valid_frames ← Ø
5.       for each frame f in p
6.          is_valid ← true
7.          for each feature value v in f
8.             if v < 0.0 or v > 5.0
9.                is_valid ← false
10.               break
11.         if is_valid
12.            valid_frames ← valid_frames ∪ {f}
13.      if valid_frames ≠ Ø
14.         D' ← D' ∪ {(p, valid_frames)}
15.   return D'
```

`is_valid`. If any feature value is outside the valid range, `is_valid` is set to false, and the inner loop breaks. If `is_valid` is true after checking all feature values in `f`, the frame `f` is added to the set `valid_frames`. After checking all frames for the participant `p`, if `valid_frames` is not empty (i.e., there are valid frames), a tuple (`p`, `valid_frames`) is added to the filtered dataset `D'`. After checking all participants, the filtered dataset `D'` is returned.

Data filtering is also applied to the eye gaze data according to the data value requirements. Eye-gaze data is eliminated if there are frames in which one of the features stores a value of 0 or -1.

b. Data Normalization

Normalization of data may be required to ensure consistent scale across features, to improve numerical stability and algorithm convergence, to improve interpretability, and to prevent the dominance of larger-scale features in distance or similarity calculations in certain machine learning methods.

Normalising AU intensity data with 14 AU features, where each feature has a value in the range of 0.0 to 5.0, and mapping it to the range of 0.0 to 1.0, can use the min-max normalisation formula. The process of normalizing AUs data is performed according to Algorithm 2.

The algorithm initializes empty variable : `normalized_data_features` (to store the normalized data), `min_values` (to store the minimum values of each participant), and `max_values` (to store the maximum values of each participant). Compute the global minimum value `global_min` from `min_values`, and the global maximum value `global_max` from `max_values`. For each participant `p` in `data_features`, compute normalize the participant's data features using the formula `(p - global_min) / (global_max - global_min)`. This scales each value in `p` to fall within the range [0.0, 1.0]. Add the normalized participant data features to `normalized_data_features`. After checking all participants, the `normalized_data_features` is returned.

Algorithm 2:Normalized Data Features
Input: Data Features where each feature value is in the range [0.0, 5.0].
Output: normalized data features containing values within the range [0.0, 1.0]).

```
1.  ALGORITHM NormalizeDataFeatures(data_features)
2.  normalized_data_features ← Ø
3.  min_values ← Ø
4.  max_values ← Ø
5.  for each participant p in data_features
6.      participant_min ← np.min(p, axis=(0, 1, 2))
7.      participant_max ← np.max(p, axis=(0, 1, 2))
8.      min_values ← min_values ∪ {participant_min}
9.      max_values ← max_values ∪ {participant_max}
10. global_min ← np.min(min_values)
11. global_max ← np.max(max_values)
12. for each participant p in data_features
13.     normalized_participant ← (p - global_min) /
        (global_max - global_min)
14.     normalized_data_features ←
        normalized_data_features ∪
        {normalized_participant}
15. return normalized_data_features
```

If x is defined as the feature value in the original range [0.0, 5.0], while the normalized value x' in the range [0.0, 1.0] can be calculated using the Eq. (1):

$$x' = \frac{x - Min(X)}{Max(X) - Min(X)} \quad (1)$$

$Min(X)$and $Max(X)$ are the minimum and maximum values of the characteristic X, respectively. In order to map the normalised value x' to the desired range [0.0, 0.1], the following transformation is applied:

$$x'' = 0.1 \times x' \quad (2)$$

Normalise the AU intensity values from the range [0.0, 5.0] to the range [0.0, 0.1] by combining Eq. (1,2):

$$x'' = 0.1 \times \frac{x - Min(X)}{Max(X) - Min(X)} \quad (3)$$

In Eq. (3), x is the original feature value in the range [0-0, 0-5], $Min(X)$ and $Max(X)$ are the minimum and maximum values of feature X respectively, and x'' is the standardised value in the desired range [0-0, 0-1]. This formula first applies min-max normalisation to scale the feature values to the range [0.0, 1.0]. It then multiplies the normalised values by 0.1 to map them to the desired range [0.0, 0.1].

c. Data Windowing

The next step is to prepare the input data using data windowing. This process is performed on the As Intensity and Eye Gaze data. Overlapping windows are created from the data set based on the specified window size and step. By sliding a window of fixed size (100) over the data, advancing by a fixed step (50) each time, and collecting these windows in a list. The process of windowing AUs data is performed according to Algorithm 3.

Algorithm 3: Data Windowing AU
Input: Data, Window Size, Step Size
Output: A windowing dataset D containing data windowing in format (100,14) with every 50 step frames

1. **ALGORITHM** DataWindowingAU(data, window_size, step_size)
2. windows_list ← empty list
3. num_windows ← (length of data - window_size) // step_size + 1
4. for i from 0 to num_windows - 1 do
5. start_index ← i * step_size
6. end_index ← start_index + window_size is_valid ← true
7. window ← data[start_index to end_index - 1]
8. append window to windows_list
9. end for
10. data ← D
11. window_size ← 100 step_size ← 50
12. sliding_windows ← DataWindowingAU(data, window_size, step_size,1)
13. D ← convert sliding_windows to D

The algorithm initializes an empty list `windows_list` to store the windows data. Compute `num_windows` (number of windows) to ensure getting all possible windows given the window size and step size. For each iteration `i`: compute the `start_index` as `i * step_size`, compute the `end_index` as `start_index + window_size`, extract the window of data from `start_index` to `end_index - 1`, and append the extracted window to `windows_list`. After the loop ends, the list `windows_list` contains all the windows. Set `window_size` to 100 and `step_size` to 50 (line 11). Call the `DataWindowingAU` function with the data and the new window size and step size. Convert the resulting `sliding_windows` into the format `D`.
The windowing format is adapted to each data matrix:
1) AU intensity
An overlapping window of 100 rows of data with 14 AUs features and a step size of 50 rows is created refers Eq. (4,5). The creation of overlapping windows using a sliding window approach defined as:

$$A = \{a_1, a_2, a_3, \dots, a_{14}\} \quad (4)$$

A represents feature set, $X = \{A_i\}$ with $i = 1 \cdots N$ where N is number of frames. We implemented data windowing:

$$X_j = \{A_{j\times s}, A_{j\times s+1}, \cdots, A_{j\times s+m}\} \quad (5)$$

the j-th window starting at row $j \times s$, with s is steps, m is window width, and $A_{j\times s}, A_{j\times s+1}, \cdots, A_{j\times s+m} \in X$.

2) Eye Gaze
An overlapping window of 100 rows of data with 12 eye gaze features and a step size of 50 rows is created refers Eq. (6,7). The creation of overlapping windows using a sliding window approach defined as:

$$G = \{g_1, g_2, g_3, \dots, g_{14}\} \quad (6)$$

G represents feature set, $Y = \{G_i\}$ with $i = 1 \cdots N$ where N is number of frames. We implemented data windowing:

$$Y_j = \{G_{j\times s}, G_{j\times s+1}, \cdots, G_{j\times s+m}\} \quad (7)$$

the j-th window starting at row $j \times s$, with s is steps, m is window width, and $G_{j\times s}, G_{j\times s+1}, \cdots, G_{j\times s+m} \in Y$.

Multi-Input CNN-WoPL Method
The method used in this proposed framework is called Multi-Input CNN-WoPL, which is a CNN-based method without a pooling layer, using 2 input features: AU intensity and eye gaze. The structure of the Multi-Input CNN-WoPL method does not use a pooling layer, as it eliminates important features from the AU intensity and gaze direction, which have an impact on the detection results [Singh 2020]. The CNN-based method used consists of:
a) Convolution Layer
A convolution layer analyses the input feature map with a convolutional filter to produce the output feature map. The convolutional layer analyses the input feature map using a convolutional filter, which involves sliding the filter over the input feature map, multiplying the filter by the corresponding region of the input at each point, and combining the results to produce a single value in the output feature map. The output size is calculated as follows:

$$OutputCov = \frac{f+2p-d}{s} + 1 \quad (8)$$

Where f is the number of filters, *p* is the amount of padding, *d* is the filter size and *s* is the stride.

b) Activation Function
Mapping input to output is the core function of all activation functions in all types of neural

networks. The input is determined by computing the weighted sum of the neurons' input and bias, if any. The activation function determines whether a neuron will fire on a given input by producing the appropriate output.

- ReLU

In the context of CNNs, this is the most commonly used function. It has a lower computation time than the others and converts all input values to positive numbers. Its mathematical representation is shown in Eq. (9).

$$f(x)_{ReLU} = \max(0, x) \quad (9)$$

- Sigmoid

A real number is used to input this function. Output will be limited between zero and one. The sigmoid curve has an S-shape and is mathematically represented by Eq. (10).

$$f(x)_{sigm} = \frac{1}{1-e^{-x}} \quad (10)$$

c) Dropout Layer

During training, the dropout layer randomly drops (sets to zero) a fraction of the neurons' activations. This contributes to the development of a more resilient and generalized network. The dropout process formula is represented by Eq. (11).

$$Output_{drop} = Input \odot Mask \quad (11)$$

d) Flatten Layer

A flattening plane transforms an input tensor from a 2D image or 3D volume into a single dimensional vector. The flattening layer equation:

$$Output_{flatten} = Input_{reshaped} \quad (12)$$

e) Fully Connected Layer

Fully connected layer (dense) is used to connect each neuron from the previous layer to each neuron in the current layer. It is used as a CNN classifier. The fully connected layer equation:

$$Output_{FC} = Activation\left(\sum_{i=1}^{n_{in}} Input\_vc_i \times Weights_i + Biases_{out}\right) \quad (13)$$

Model Prediction Evaluation

In k-fold cross-validation, each dataset is divided into k-folds, with k-1 folds used for training samples and the remaining folds used for test samples. This process is repeated k times so that all subsets are tested, with equation:

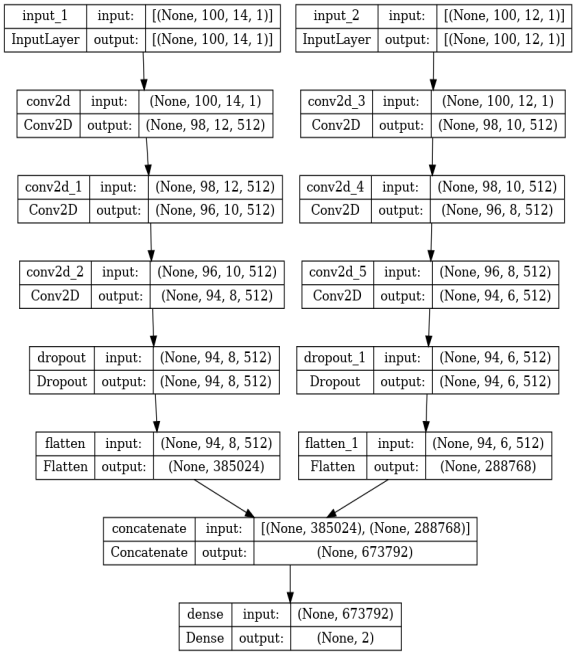

Figure 2. Multi-Input CNN-WoPL plot model

$$Pred\_Ev = \frac{1}{k}\sum_{i=1}^{k} Accuracy^k \quad (14)$$

Classification Performance Evaluation

By examining the performance of classification-based machine learning models, the confusion matrix is a tool for predictive analysis in machine learning. Furthermore, the confusion matrix is a simplified table of the number of correct and false predictions made by a classifying machine (or classification model) for a binary classification task. Each classification performance measurement can be defined as follows:

Accuracy is the ratio of correctly classified example data to the total number of data samples. Accuracy is one of the most commonly used measures of classifying performance. [23], with equation:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

Precision Precision represents the proportion of correctly classified positive data samples out of the total number of positive prediction data samples. [23], with equation:

$$Precision = \frac{TP}{TP+FP} \quad (16)$$

Recall represent positive samples data that are correctly classified to the total number of positive samples data and are estimated [23], with equation:

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

F1 score represents the average harmonic of precision and recall. The equation:

$$F1 - Score = \frac{2\times(Recall\times Precision)}{Recall+Precision} \quad (18)$$

RESULT AND DISCUSSION

Data Variation

The data is stored in a 4-dimensional array for features and a 2-dimensional array for labels: Action Unit ((19548, 100, 14, 1), (19548, 2)), and Eye Gaze ((19548, 100, 12, 1), (19548, 2)). With these large dimensions, the model will be able to capture local details in each feature related to depression. In the Action Unit intensity feature, small changes in the face (eyebrows or mouth) are very important because they will affect the expression of facial emotions related to depression. Likewise, with the Eye Gaze coordinate feature, maintaining spatial details is also important to highlight small but meaningful eye gaze directions related to depression. A large spatial dimension also helps to maintain the precision of the eye position.

Table 4. Detail of data variation and data split

Data		Action Units (AUs)			Eye Gaze		
		Training	Validation	Testing	Training	Validation	Testing
25% (A)	1	90% (4388, 100, 14, 1), (4388, 2)	5% (244, 100, 14, 1), (244, 2)	5% (244, 100, 14, 1), (244, 2)	90% (4388, 100, 12, 1), (4388, 2)	5% (244, 100, 12, 1), (244, 2)	5% (244, 100, 12, 1), (244, 2)
	2	80% (3900, 100, 14, 1), (3900, 2)	10% (488, 100, 14, 1), (488, 2)	10% (488, 100, 14, 1), (488, 2)	80% (3900, 100, 12, 1), (3900, 2)	10% (488, 100, 12, 1), (488, 2)	10% (488, 100, 12, 1), (488, 2)
	3	70% (3413, 100, 14, 1), (3413, 2)	15% (731, 100, 14, 1), (731, 2)	15% (732, 100, 14, 1), (731, 2)	70% (3413, 100, 12, 1), (3413, 2)	15% (731 100, 12, 1), (731, 2)	15% (732, 100, 12, 1), 7328, 2)
	4	60% (2925, 100, 14, 1), (2925, 2)	20% (975, 100, 14, 1), (975, 2)	20% (976, 100, 14, 1), (976, 2)	60% (2925, 100, 12, 1), (2925, 2)	20% (975, 100, 12, 1), (975, 2)	20% (976, 100, 12, 1), (976, 2)
50% (B)	1	90% (8788, 100, 14, 1), (8788, 2)	5% (488, 100, 14, 1), (488, 2)	5% (489, 100, 14, 1), (489, 2)	90% (8788, 100, 12, 1), (8788, 2)	5% (488, 100, 12, 1), (488, 2)	5% (489, 100, 12, 1), (489, 2)
	2	80% (7812, 100, 14, 1), (7812, 2)	10% (976, 100, 14, 1), (976, 2)	10% (977, 100, 14, 1), (977, 2)	80% (7812, 100, 12, 1), (7812, 2)	10% (976, 100, 12, 1), (976, 2)	10% (977, 100, 12, 1), (977, 2)
	3	70% (6835, 100, 14, 1), (6835, 2)	15% (1465, 100, 14, 1), (1465, 2)	15% (1465, 100, 14, 1), (1465, 2)	70% (6835, 100, 12, 1), (6835, 2)	15% (1465, 100, 12, 1), (1465, 2)	15% (1465, 100, 12, 1), (1465, 2)
	4	60% (5859, 100, 14, 1), (5859, 2)	20% (1953, 100, 14, 1), (1953, 2)	20% (1465, 100, 14, 1), (1465, 2)	60% (5859, 100, 12, 1), (5859, 2)	20% (1953, 100, 12, 1), (1953, 2)	20% (1465, 100, 12, 1), (1465, 2)
75% (C)	1	90% (13194, 100, 14, 1), (13194, 2)	5% (733, 100, 14, 1), (733, 2)	5% (733, 100, 14, 1), (733, 2)	90% (13194, 100, 12, 1), (13194, 2)	5% (733, 100, 12, 1), (733, 2)	5% (733, 100, 12, 1), (733, 2)
	2	80% (11728, 100, 14, 1), (11728, 2)	10% (1466, 100, 14, 1), (1466, 2)	10% (1466, 100, 14, 1), (1466, 2)	80% (11728, 100, 12, 1), (11728, 2)	10% (1466, 100, 12, 1), (1466, 2)	10% (1466, 100, 12, 1), (1466, 2)
	3	70% (10262, 100, 14, 1), (10262, 2)	15% (2199, 100, 14, 1), (2199, 2)	15% (2199, 100, 14, 1), (2199, 2)	70% (10262, 100, 12, 1), (10262, 2)	15% (2199, 100, 12, 1), (2199, 2)	15% (2199, 100, 12, 1), (2199, 2)
	4	60% (8796, 100, 14, 1), (8796, 2)	20% (2932, 100, 14, 1), (2932, 2)	20% (2932, 100, 14, 1), (2932, 2)	60% (8796, 100, 12, 1), (8796, 2)	20% (2932, 100, 12, 1), (2932, 2)	20% (2932, 100, 12, 1), (2932, 2)
100% (D)	1	90% (17593, 100, 14, 1), (17593, 2)	5% (977, 100, 14, 1), (977, 2)	5% (978, 100, 14, 1), (978, 2)	90% (17593, 100, 12, 1), (17593, 2)	5% (977, 100, 12, 1), (977, 2)	5% (978, 100, 12, 1), (978, 2)
	2	80% (15638, 100, 14, 1), (15638, 2)	10% (1955, 100, 14, 1), (1955, 2)	10% (1955, 100, 14, 1), (1955, 2)	80% (15638, 100, 12, 1), (15638, 2)	10% (1955, 100, 12, 1), (1955, 2)	10% (1955, 100, 12, 1), (1955, 2)
	3	70% (13683, 100, 14, 1), (13683, 2)	15% (2932, 100, 14, 1), (2932, 2)	15% (2932, 100, 14, 1), (2932, 2)	70% (13683, 100, 12, 1), (13683, 2)	15% (2932, 100, 12, 1), (2932, 2)	15% (2932, 100, 12, 1), (2932, 2)
	4	60% (11728, 100, 14, 1), (11728, 2)	20% (3910, 100, 14, 1), (3910, 2)	20% (3910, 100, 14, 1), (3910, 2)	60% (11728, 100, 12, 1), (11728, 2)	20% (3910, 100, 12, 1), (3910, 2)	20% (3910, 100, 12, 1), (3910, 2)

Using a Pooling layer will make the spatial dimension smaller and capture more abstract features, thus not noticing small changes in the face and eye gaze related to depression. Not using a pooling layer is beneficial for preserving small details, which are important for tasks such as detecting subtle expressions and eye gaze direction related to depression.

Table 4 illustrates the creation of data variations using the percentage of data used as input: 25% (A), 50% (B), 75% (C) and 100% (D). In addition, for each percentage of data used, the data is split into training data, validation data and test data with variations: (1) 90%:5%:5%, (2) 80%:10%:10%, (3) 70%:15%:15% and (4) 60%:20%:20%. The table also shows how much data is used in each data variation. The total AU intensity data is (19548,100,14,1), then Data A1 represents the use of 25% AU intensity data (4876,100,14,1) which is split for Training Data:Validation Data:Testing Data by 90%:5%:5%, so the data split becomes (4388,100,14,1):(244,100,14,1):(244,100,14,1).

Model Multi-Input CNN-WoPL

The Multi-Input CNN-WoPL method has 2 branches based on a CNN without a pooling layer. Each branch consists of 3 x convolutional layers each consisting of 512 filter units, where each layer has a kernel size of 3 × 3, see Eq. (8), applies L2 kernel regularisation (0.001), and uses the ReLu activation function, see Eq. (9). The dropout layer is added to anticipate overfitting, with a dropout rate of 0.8, see Eq. (11). 1 x the Flatten layer, which converts the feature map resulting from the Convolution layer into vector form, see Eq. (12). The Flatten layer outputs from each branch are then combined using the Concatenate function, and the output becomes the input for the output layer (Dense). Then, the output layer (Dense) with the number of neurons 2 (according to the output class) uses the sigmoid activation functions to classify according to the expected output (Depression/Non-Depression), see Eq. (13). Multi-Input CNN-WoPL plot model as shown in Fig. 2.

The resulting shape output from each layer is as shown in Fig. 3. The structure of the Multi-Input CNN-WoPL method does not use Pooling Layer because it eliminates essential features from AU intensity and eye gaze which affects the classification results.

Experiments

We conducted the experiments based on Python programming using Jupyter Notebook on NVIDIA DGX A100 (8 Tensor Core GPU, 320 GB GPU Memory, 1 TB Sistem Memory). The

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 100, 14, 1)]	0	
input_2 (InputLayer)	[(None, 100, 12, 1)]	0	
conv2d (Conv2D)	(None, 98, 12, 512)	5120	input_1[0][0]
conv2d_3 (Conv2D)	(None, 98, 10, 512)	5120	input_2[0][0]
conv2d_1 (Conv2D)	(None, 96, 10, 512)	2359808	conv2d[0][0]
conv2d_4 (Conv2D)	(None, 96, 8, 512)	2359808	conv2d_3[0][0]
conv2d_2 (Conv2D)	(None, 94, 8, 512)	2359808	conv2d_1[0][0]
conv2d_5 (Conv2D)	(None, 94, 6, 512)	2359808	conv2d_4[0][0]
dropout (Dropout)	(None, 94, 8, 512)	0	conv2d_2[0][0]
dropout_1 (Dropout)	(None, 94, 6, 512)	0	conv2d_5[0][0]
flatten (Flatten)	(None, 385024)	0	dropout[0][0]
flatten_1 (Flatten)	(None, 288768)	0	dropout_1[0][0]
concatenate (Concatenate)	(None, 673792)	0	flatten[0][0] flatten_1[0][0]
dense (Dense)	(None, 2)	1347586	concatenate[0][0]

Total params: 10,797,058
Trainable params: 10,797,058
Non-trainable params: 0

Figure 3. The resulting shape otput model Multi-Input CNN-WoPL

experiments were carried out with all the data variations, with the data split according to the Table 4. The optimising parameters used are based on our previous studies [43], including Number of Filters, Activation Function, Learning Rate and Batch Size. The best parameter values are generated from hyper-parameter tuning using the GridSearch algorithm. From the tuning parameter results, The Multi-Input CNN-WoPL was used to train the data, using the following parameters: Number of Filters=512, Activation Function=Adam, Learning Rate=0.0001 and Batch Size=64 with 25 epochs. During the data training process, each prediction result of the Multi-Input CNN-WoPL method was evaluated using the K-fold cross-validation with a value of K = 5.

Using 25% data variation, the A1 data (see Table 4) produced average values during data training of Loss = 0.369 and Accuracy = 0.993 for training, Loss = 0.491 and Accuracy = 0.950 for validation, and Loss = 0.481 and Accuracy = 0.947 for testing. A2 data produced average values during data training with Loss = 0.410 and Accuracy = 0.983 for training, while Loss = 0.574 and Accuracy = 0.914 for validation and Loss = 0.646 and Accuracy = 0.877 for testing. A3 data produced average values during data training with Loss = 0.421 and Accuracy = 0.985 for training, while Loss = 0.606 and Accuracy = 0.908 for validation and Loss = 0.755 and Accuracy = 0.866 for testing. A4 data produced average values during data training with Loss = 0.505 and

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

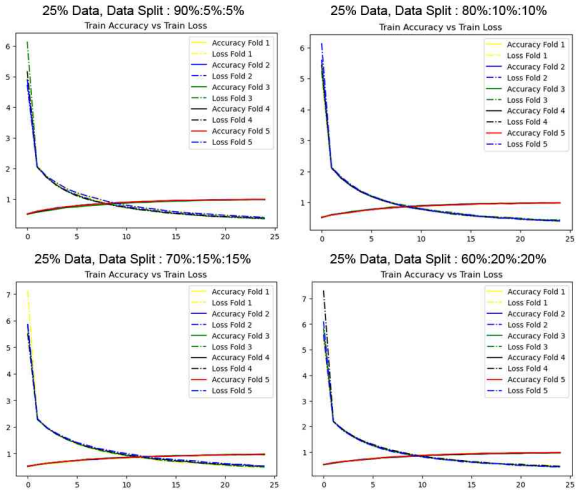

Figure 4. Training with 25% data.

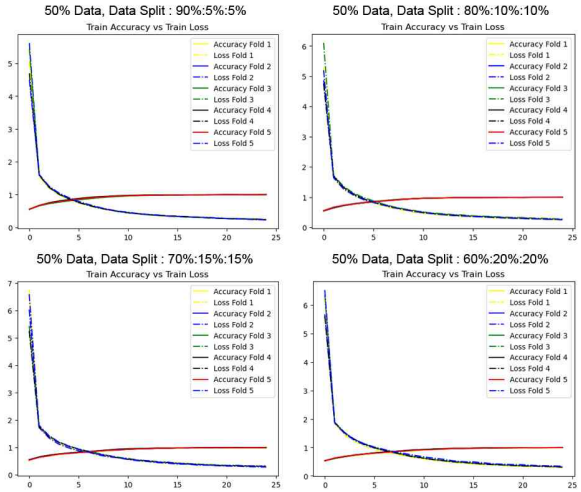

Figure 5. Training with 50% data.

Accuracy = 0.974 for training, while Loss = 0.766 and Accuracy = 0.829 for validation and Loss = 0.848 and Accuracy = 0.850 for testing, according to Table 5. The resulting graph of each process during data training for 25% data variation, as shown in Fig. 4.

Using 50% data variation, the B1 data (see Table 4) produced average values during data training of Loss = 0.232 and Accuracy = 0.994 for training, Loss = 0.252 and Accuracy = 0.986 for validation, and Loss = 0.242 and Accuracy = 0.984 for testing. B2 data produced average values during data training with Loss = 0.257 and Accuracy = 0.995 for training, while Loss = 0.433 and Accuracy = 0.981 for validation and Loss = 0.282 and Accuracy = 0.980 for testing. B3 data produced average values during data training with Loss = 0.287 and Accuracy = 0.994 for training, while Loss = 0.349 and Accuracy = 0.894 for validation and Loss = 0.327 and Accuracy = 0.978 for testing. B4 data produced average values during data training with Loss = 0.309 and Accuracy = 0.993 for training, while Loss = 0.256 and Accuracy = 0.973 for validation and Loss = 0.376 and Accuracy = 0.969 for testing, according to Table 5. The resulting graph of each process during data training for 50% data variation, as shown in Fig. 5.

Table 5. Result of the experiments

Data Variant	Training		Validation		Testing	
	Loss	Acc	Loss	Acc	Loss	Acc
A1	0.369	0.993	0.491	0,950	0,481	0,947
A2	0.410	0.983	0.574	0,914	0,646	0,877
A3	0.421	0.985	0.606	0,908	0,755	0,866
A4	0.505	0.974	0.766	0,829	0,848	0,850
B1	0.232	0.994	0.252	0,986	0,242	0,984
B2	0.257	0.995	0.433	0,981	0,282	0,980
B3	0.287	0.994	0.349	0,849	0,327	0,978
B4	0.309	0.993	0.356	0,973	0,376	0,969
C1	0.171	0.996	0.195	0,987	0,200	0,985
C2	0.190	0.997	0.208	0,989	0,207	0,988
C3	0.208	0.996	0.229	0,988	0,218	0,989
C4	0.232	0.997	0.259	0,984	0,255	0,986
D1	0.407	0.957	0.397	0,955	0,157	0,994
D2	**0.158**	**0.998**	**0.169**	**0,994**	**0,161**	**0,994**
D3	0.279	0.981	0.305	0,973	0,189	0,992
D4	0.197	0.998	0.214	0,992	0,208	0,992

Using 75% data variation, the C1 data (see Table 4) produced average values during data training of Loss = 0.171 and Accuracy = 0.996 for training, Loss = 0.195 and Accuracy = 0.987 for validation, and Loss = 0.200 and Accuracy = 0.985 for testing. C2 data produced average values during data training with Loss = 0.190 and Accuracy = 0.997 for training, while Loss = 0.208 and Accuracy = 0.989 for validation and Loss =

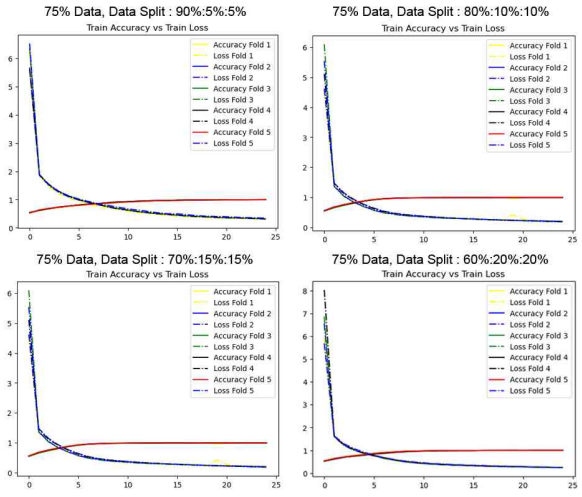

Figure 6. Training with 75% data.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

0.218 and Accuracy = 0.988 for testing. C3 data produced average values during data training with Loss = 0.208 and Accuracy = 0.996 for training, while Loss = 0.229 and Accuracy = 0.988 for validation and Loss = 0.218 and Accuracy = 0.989 for testing. C4 data produced average values during data training with Loss = 0.232 and Accuracy = 0.997 for training, while Loss = 0.259 and Accuracy = 0.984 for validation and Loss = 0.255 and Accuracy = 0.986 for testing, according to Table 5. The resulting graph of each process during data training for 75% data variation, as shown in Fig. 6.

Using 100% data variation, the D1 data (see Table 3) produced average values during data training of Loss = 0.407 and Accuracy = 0.957 for training, Loss = 0.397 and Accuracy = 0.955 for validation, and Loss = 0.157 and Accuracy = 0.994 for testing. D2 data produced average values during data training with Loss = 0.158 and Accuracy = 0.998 for training, while Loss = 0.169 and Accuracy = 0.994 for validation and Loss = 0.161 and Accuracy = 0.994 for testing. D3 data produced average values during data training with Loss = 0.279 and Accuracy = 0.981 for training, while Loss = 0.305 and Accuracy = 0.973 for validation and Loss = 0.189 and Accuracy = 0.992 for testing. D4 data produced average values during data training with Loss = 0.197 and Accuracy = 0.998 for training, while Loss = 0.214 and Accuracy = 0.992 for validation and Loss = 0.208 and Accuracy = 0.992 for testing, according to Table 4. The resulting graph of each process during data training for 100% data variation, as shown in Fig. 7

The performance of the proposed framework achieves a good level of accuracy. By using 5-Fold Cross-Validation, see Eq. (14), it is able to produce an average Loss and Accuracy

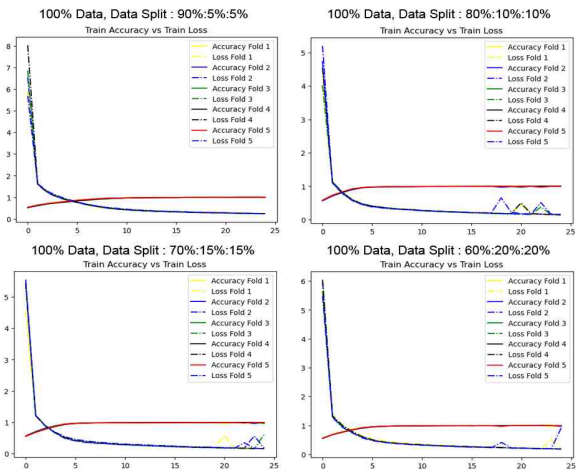

Figure 7. Training with 100% data.

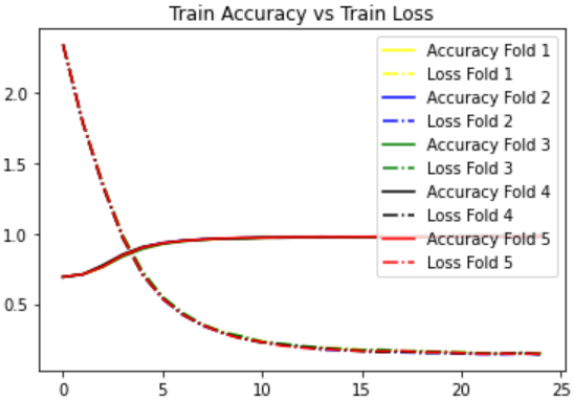

Figure 8. 5-fold cross-validation graph on the performance of the proposed framework

value of Training (Loss = 0.158, Accuracy = 0.998), Validation (Loss = 0.169, Accuracy = 0.994) and Testing (Loss = 0.161, Accuracy = 0.994). A graph during the training process on the proposed framework using 5-fold cross-validation with an epoch value = 50 and a processing speed of 18ms for each step, as shown in Fig. 8.

Result

In the experiments carried out, each data variation (A, B, C and D) gave the best results, as shown in Fig. 9. From these experiments, the best value are Loss = 0.158 and Accuracy = 0.998 for Training, Loss = 0.169 and Accuracy = 0.994 for Validation, and Loss = 0.161 and Accuracy = 0.994 for Test on data variation D2, according to Table 5. Data variation D2 consists of 80% Training Data, 10% Validation Data, 10% Testing Data. The size of the data is Training Data (Feature: (13683,100,14,1); Label: (13683,2)), Validation Data (Features: (2932,100,14,1); Labels: (2932,2)), and Training Data (Features: (2932,100,14,1); Labels: (2932,2)).

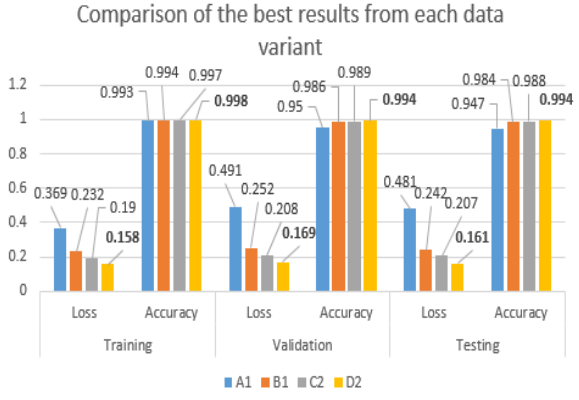

Figure 9. Comparison of the best results from each data variant

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Performance Evaluation of The Proposed Framework

The performance of the proposed framework is assessed using the Confusion Matrix. True Positive = 1064 and True Negative = 880 are the results of calculating the performance of the method using a confusion matrix. When TP and TN are added, the result is 1944 or 0.994 from the total test data of 1955 samples. This means that the accuracy results in the test are the same as the results in the confusion matrix calculations.

Based on the results of the calculation of TP, TN, FP and FN, the calculation of accuracy,

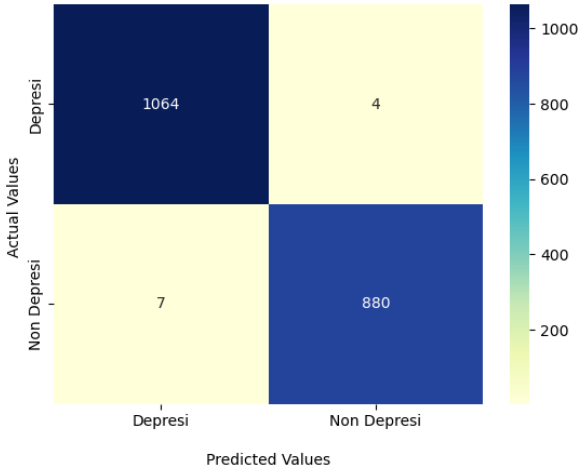

Figure 10. Result of Performance evaluation with Confusion matrix

precision, recall and F1 score are as follows:

1. Accuracy = 0.994; the resulting accuracy value is equal to the accuracy value of the proposed test method, see Eq. (15).
2. Precision = 0.995; the precision values are close to the ideal values (1.0) to compare true positives (TP) and the amount of data predicted to be positive, see Eq. (16).
3. Recall = 0.992; the precision value is close to the ideal value (1.0) for the comparison between TP and the number of positive data, see Eq. (17).
4. F1-Score = 0.993, the best F1-Score value is 1.0 and the worst is 0, see Eq. (18).

The performance of the proposed method was evaluated using the Confusion Matrix, which resulted in an Accuracy Score of 0.994 according to the test results of the proposed method. In addition, the value of F1 score = 0.993 indicates that the proposed method has precision, recall and good performance.

Comparing with Adding a Pooling Layer to The Proposed Framework and Other Method

Table 6. Comparison Method

Method	Training		Validation		Testing	
	Loss	Acc	Loss	Acc	Loss	Acc
M1*)	**0.158**	**0.998**	**0.169**	**0,994**	**0,161**	**0,994**
M2*)	0.186	0.989	0.298	0,988	0,232	0,987
Song [31]	0.102	0.971	0.664	0.732	0.646	0.748

*) M1 : Multi-Input CNN Without Pooling Layer
M2 : Multi-Input CNN With Pooling Layer

We also added a performance comparison by adding a pooling layer to the methods in the proposed framework. The comparison is done by using the D2 variation of the data, which results in the best performance. We obtained the results of Training (Loss = 0.186, Accuracy = 0.989), Validation (Loss = 0.298, Accuracy = 0.988) and Testing (Loss = 0.232, Accuracy = 0.987).
It is also compared with a method from related research. Song [2018] used CNN method with an architecture consisting of : Convolutional Layer 64, Batch Normalization Layer, ReLU Layer → Convolutional Layer 128, Batch Normalization Layer, ReLU Layer → Convolutional Layer 64, Batch Normalization Layer, ReLU Layer → MaxPooling Layer → Fully Connected Layer → LogSoftMax → Result. The method resulted in Loss=0.646 and Accuracy=0.748. Compairing results are still below the classification results of proposed framework that do not use the pooling layer, as shown in Table 6.

Robustness test of the proposed framework

The robustness of the proposed framework has been tested against several challenges, including:

- Noisy data
  The robustness to noisy data is tested by adding noise to the data using Gaussian Noise with the main parameters mean = 0 and standard deviation = 0.05 and having the same size as the original data. With a standard deviation of 0.05, the addition of the noise can shift the value of the original data by approximately ±0.05. The robustness of the model is proven, the data with added noise gives a loss value = 0.059, accuracy value = 0.980 and F1 Score = 0.976, which is not far away from the accuracy results obtained using the original data (loss = 0.161, accuracy = 0.994 and F1-Score = 0.993).
- Data Augmentation
  Data augmentation testing is performed using two augmentation techniques: shift and scale. In the AU intensity data, shift augmentation with the parameter shift_value=0.05 means

Table 7. Robustness test of the proposed framework

Strategy		Loss	Accuracy	Precision	Recall	F1-Score
Noisy Data		0.059	0.980	0.996	0.959	0.976
Data Augmentation	Shift	0.192	0.993	0.994	0.993	0.993
	Scale	0.044	0.983	0.994	0.968	0.981
Missing Data		0.046	0.983	0.981	0.982	0.981
Proposed Framework		**0.161**	**0.994**	**0.994**	**0.994**	**0.993**

that a value of ±0.05 is added or subtracted from the original AU intensity value to create variation, as if the facial expression is subtly changing. In the case of gaze coordinate data, shift augmentation with shift_value=0.02 means that the augmentation shifts the gaze position by a value of ±0.02 in the gaze coordinate. Scale augmentation is performed with the parameter scale_factor=1.05. In the AU intensity data, the AU intensity is scaled 5% higher or lower. In the gaze coordinate, it will increase or decrease the gaze distance in space by 5%.

The robustness of the model with the shift augmentation technique is proven by the results of the loss value = 0.192, accuracy value = 0.993 and F1-Score = 0.993. Meanwhile robustness of the model with the scale augmentation technique is proven by the results of the loss value = 0.044, accuracy value = 0.983 and F1-Score = 0.981. Both augmentation techniques provide results that are not far from the accuracy results obtained with the original data (loss = 0.161, accuracy = 0.994, and F1-Score = 0.993).

- Missing Data

  The robustness test uses Missing Data with a drop rate=0.1, which means that the drop rate or percentage of data to be deleted in AU intensity data and eye gaze coordinates from the original data is 10%. The test using the Missing Data technique has proven the robustness of the model by producing a loss value = 0.046, accuracy value = 0.983 and F1-Score = 0.981 which are not far from the accuracy using the original data (loss = 0.161, accuracy = 0.994, and and F1-Score = 0.993).

Discussion

Action Units represent important depression-related local changes in specific parts of the face, like the brows, mouth, or eyes. Eye gaze coordinates are spatial features that are also very important in the detection of depression, as small changes in the direction of gaze can be very significant. Pooling layers are often used in CNNs in order to reduce the spatial dimension of the feature map. The reduced spatial dimension can result in the loss of some local details, which can affect the results of depression detection. By removing the pooling layer, the spatial dimension of the feature map remains larger throughout the network. This means that the model retains more spatial information of the input features, which is crucial to better capture small details in action unit intensity and gaze movements associated with depression. Large amounts of important information can be lost due to the pooling layer's ability to reduce data dimensions, which can lead to data discontinuity [40]. Thus, the depression detection performance of the model without the pooling layer is improved.

We also add a performance comparison of the proposed framework with methods from related research and adding a pooling layer to the method in the proposed framework. Using the data variation with the best classification result (D2), which represents 100% data usage with a split of training data: validation data: test data of 80%:10%:10%, we obtained the results of Training (Loss = 0.186, Accuracy = 0.989), Validation (Loss = 0.298, Accuracy = 0.988) and Testing (Loss = 0.232, Accuracy = 0.987). These results are still below the classification results of methods that do not use the pooling layer, as shown in Table 5. Tests were also conducted on the model's robustness to data noise, data variation through augmentation with shifting and scaling techniques, and missing data. The test results show that the model has good robustness, as evidenced by the test results shown in Table 7.

CONCLUSION

In this study, we propose a framework for depression detection using action unit intensity and gaze features extracted from facial expressions. From various experiments conducted with different data variations: the amount of data and the split of data for training, validation and testing, our proposed framework gives excellent results. Our proposed framework

achieved the best results with an accuracy value of 0.994 and a loss value of 0.161. Furthermore, the proposed framework is shown to have good robustness against data noise, data variation, and missing data.This automatic depression detection is expected to be used to detect depression symptoms earlier, faster and more accurately, so that it can provide appropriate treatment to reduce the prevalence of advanced depression, which has a high suicide risk rate. The next research is to predict the severity of depression using multimodal features: AU intensity, eye gaze and facial landmarks. The use of different features is expected to provide prediction results with better and more stable accuracy.

ACKNOWLEDGMENT

We would like to thank the Indonesian Ministry of Research, Technology and Higher Education's BPPDN Scholarship Programme 2019.

REFERENCES

[1] World Health Organization, *Depression and Other Common Mental Disorders: Global Health Estimates*. Geneva: World Health Organization, Licence: CCBY-NC-SA 3.0 IGO, 2017.

[2] M. Nordentoft, P. B. Mortensen, and C. B. Pedersen, 'Absolute Risk of Suicide After First Hospital Contact in Mental Disorder', *ARCH GEN PSYCHIATRY*, vol. 68, no. 10, 2011.

[3] M. Gili *et al.*, 'Mental disorders as risk factors for suicidal behavior in young people: A meta-analysis and systematic review of longitudinal studies', *Journal of Affective Disorders*, vol. 245, pp. 152–162, Feb. 2019, doi: 10.1016/j.jad.2018.10.115.

[4] M. Dong *et al.*, 'Prevalence of suicide attempt in individuals with major depressive disorder: a meta-analysis of observational surveys', *Psychol. Med.*, vol. 49, no. 10, pp. 1691–1704, Jul. 2019, doi: 10.1017/S0033291718002301.

[5] H. Cai *et al.*, 'Prevalence of suicidal ideation and planning in patients with major depressive disorder: A meta-analysis of observation studies', *Journal of Affective Disorders*, vol. 293, pp. 148–158, Oct. 2021, doi: 10.1016/j.jad.2021.05.115.

[6] M. K. Nock, I. Hwang, N. A. Sampson, and R. C. Kessler, 'Mental disorders, comorbidity and suicidal behavior: Results from the National Comorbidity Survey Replication', *Mol Psychiatry*, vol. 15, no. 8, pp. 868–876, Aug. 2010, doi: 10.1038/mp.2009.29.

[7] J. K. Hohls, H.-H. König, E. Quirke, and A. Hajek, 'Anxiety, Depression and Quality of Life—A Systematic Review of Evidence from Longitudinal Observational Studies', *Int. J. Environ. Res. Public Health*, 2021.

[8] H. Minnis, R. Gajwani, and D. Ougrin, 'Editorial: Early intervention and prevention of severe mental illness: A child and adolescent psychiatry perspective', *Front. Psychiatry*, vol. 13, p. 963602, Jul. 2022, doi: 10.3389/fpsyt.2022.963602.

[9] S. Saldivia, F. Cova, C. Inostroza, J. Aslan, and M. Farhang, 'Preventive and Early Treatment of Depression in Older Adults', in *Prevention and Early Treatment of Depression Through the Life Course*, V. Martínez and C. Miranda-Castillo, Eds., Cham: Springer International Publishing, 2023, pp. 167–187. doi: 10.1007/978-3-031-13029-8_9.

[10] B. Bohman, 'Clinicians' perceptions and practices of diagnostic assessment in psychiatric services', *BMC Psychiatry*, vol. 23, no. 1, p. 191, Mar. 2023, doi: 10.1186/s12888-023-04689-w.

[11] J. M. Aultman, 'Psychiatric Diagnostic Uncertainty: Challenges to Patient-Centered Care', *AMA Journal of Ethics*, vol. 18, no. 6, pp. 579–586, Jun. 2016, doi: 10.1001/journalofethics.2016.18.6.ecas2-1606.

[12] S. Song, S. Jaiswal, L. Shen, and M. Valstar, 'Spectral Representation of Behaviour Primitives for Depression Analysis', *IEEE Trans. Affective Comput.*, vol. 13, no. 2, pp. 829–844, Apr. 2022, doi: 10.1109/TAFFC.2020.2970712.

[13] K. C. D. Lacerda *et al.*, 'High depressive symptomatology reduces emotional reactions to pictures of social interaction', *Sci Rep*, vol. 14, no. 1, p. 1266, Jan. 2024, doi: 10.1038/s41598-024-51813-1.

[14] J. Joshi, R. Goecke, G. Parker, and M. Breakspear, 'Can body expressions contribute to automatic depression analysis?', in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Shanghai, China: IEEE, Apr. 2013, pp. 1–7. doi: 10.1109/FG.2013.6553796.

[15] J. M. Girard, J. F. Cohn, M. H. Mahoor, S. M. Mavadati, Z. Hammal, and D. P. Rosenwald, 'Nonverbal social withdrawal in depression: Evidence from manual and automatic analyses', *Image and Vision Computing*, vol. 32, no. 10, pp. 641–647, Oct. 2014, doi: 10.1016/j.imavis.2013.12.007.

[16] P. Ekman, W. V. Friesen, and J. Hager, *Facial action coding system: A technique for the measurement of facial movement*. The Manual on CDROM, 2002.
[17] D. Venkataraman and N. S. Parameswaran, 'Extraction of Facial Features for Depression Detection among Students', *International Journal of Pure and Applied Mathematics*, vol. 118 (7), no. Special Issue, pp. 455–463, 2018.
[18] A. Savran, B. Sankur, and M. T. Bilge, 'Regression-based intensity estimation of facial action unit', *Image and Vision Computing*, vol. 30, no. 10, pp. 774–784, Oct. 2012, doi: doi.org/10.1016/j.imavis.2011.11.008.
[19] Z. Ming, A. Bugeau, J.-L. Rouas, and T. Shochi, 'Facial Action Units intensity estimation by the fusion of features with multi-kernel Support Vector Machine', in : *Proc. of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) 2015*, Ljubljana: IEEE, May 2015, pp. 1–6. doi: 10.1109/FG.2015.7284870.
[20] Y. Li, S. M. Mavadati, M. H. Mahoor, Y. Zhao, and Q. Ji, 'Measuring the intensity of spontaneous facial action units with dynamic Bayesian network', *Pattern Recognition*, vol. 48, no. 11, pp. 3417–3427, Nov. 2015, doi: 10.1016/j.patcog.2015.04.022.
[21] H. Akbar, S. Dewi, Y. A. Rozali, L. P. Lunanta, N. Anwar, and D. Anwar, 'Exploiting Facial Action Unit in Video for Recognizing Depression using Metaheuristic and Neural Networks', in : *Proc. of the 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI) 2021*, Jakarta, Indonesia: IEEE, Oct. 2021, pp. 438–443. doi: 10.1109/ICCSAI53272.2021.9609747.
[22] Z. Jiang, S. Harati, A. Crowell, H. S. Mayberg, S. Nemati, and G. D. Clifford, 'Classifying Major Depressive Disorder and Response to Deep Brain Stimulation Over Time by Analyzing Facial Expressions', *IEEE Trans. Biomed. Eng.*, vol. 68, no. 2, pp. 664–672, Feb. 2021, doi: 10.1109/TBME.2020.3010472.
[23] C. Yu, 'Non-verbal Facial Action Units-based Automatic Depression Classification', Nov. 20, 2022, *arXiv*: arXiv:2211.10911. Accessed: Jun. 15, 2024. [Online]. Available: http://arxiv.org/abs/2211.10911
[24] M. Wen, Z. Dong, L. Zhang, B. Li, Y. Zhang, and K. Li, 'Depression and Cognitive Impairment: Current Understanding of Its Neurobiology and Diagnosis', *NDT*, vol. Volume 18, pp. 2783–2794, Nov. 2022, doi: 10.2147/NDT.S383093.
[25] T. Shu, F. Zhang, and X. Sun, 'Gaze Behavior based Depression Severity Estimation', in *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, Urumqi, China: IEEE, Aug. 2023, pp. 313–319. doi: 10.1109/PRML59573.2023.10348319.
[26] O. Kobo, A. Meltzer-Asscher, J. Berant, and T. Schonberg, 'Classification of depression tendency from gaze patterns during sentence reading', *Biomedical Signal Processing and Control*, vol. 93, p. 106015, Jul. 2024, doi: 10.1016/j.bspc.2024.106015.
[27] J. F. Cohn *et al.*, 'Detecting depression from facial actions and vocal prosody', in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, Amsterdam, Netherlands: IEEE, Sep. 2009, pp. 1–7. doi: 10.1109/ACII.2009.5349358.
[28] Q. Wang, H. Yang, and Y. Yu, 'Facial expression video analysis for depression detection in Chinese patients', *Journal of Visual Communication and Image Representation*, vol. 57, pp. 228–233, Nov. 2018, doi: 10.1016/j.jvcir.2018.11.003.
[29] N. Ghadiri, R. Samani, and F. Shahrokh, 'Integration of Text and Graph-Based Features for Depression Detection Using Visibility Graph', in *Intelligent Systems Design and Applications*, A. Abraham, S. Pllana, G. Casalino, K. Ma, and A. Bajaj, Eds., Cham: Springer Nature Switzerland, 2023, pp. 332–341.
[30] Z. Li, Z. An, W. Cheng, J. Zhou, F. Zheng, and B. Hu, 'MHA: a multimodal hierarchical attention model for depression detection in social media', *Health Inf Sci Syst*, vol. 11, no. 1, p. 6, Jan. 2023, doi: 10.1007/s13755-022-00197-5.
[31] S. Song, L. Shen, and M. Valstar, 'Human Behaviour-Based Automatic Depression Analysis Using Hand-Crafted Statistics and Deep Learned Spectral Features', in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, Xi'an: IEEE, May 2018, pp. 158–165. doi: 10.1109/FG.2018.00032.
[32] C. Chakraborty, M. Bhattacharya, S. Pal, and S.-S. Lee, 'From machine learning to deep learning: Advances of the recent data-driven paradigm shift in medicine and healthcare', *Current Research in Biotechnology*, vol. 7, p. 100164, 2024, doi: 10.1016/j.crbiot.2023.100164.

[33] L. Koushik, 'Interns@LT-EDI : Detecting Signs of Depression from Social Media Text'.
[34] X. Kong, Y. Yao, C. Wang, Y. Wang, J. Teng, and X. Qi, 'Automatic Identification of Depression Using Facial Images with Deep Convolutional Neural Network', *Med Sci Monit*, vol. 28, Jun. 2022, doi: 10.12659/MSM.936409.
[35] Z. Wang, C. Hu, W. Liu, X. Zhou, and X. Zhao, 'EEG-based high-performance depression state recognition', *Front. Neurosci.*, vol. 17, p. 1301214, Jan. 2024, doi: 10.3389/fnins.2023.1301214.
[36] A. Hassan and S. Bernadin, 'A Comprehensive Analysis of Speech Depression Recognition Systems', in *SoutheastCon 2024*, Atlanta, GA, USA: IEEE, Mar. 2024, pp. 1509–1518. doi: 10.1109/SoutheastCon52093.2024.10500078.
[37] N. Ignatiev, I. Smirnov, and M. Stankevich, 'Predicting Depression with Text, Image, and Profile Data from Social Media':, in *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods*, Online Streaming, --- Select a Country ---: SCITEPRESS - Science and Technology Publications, 2022, pp. 753–760. doi: 10.5220/0010986100003122.
[38] M. Patil, P. Mukherji, and V. Wadhai, 'A novel hybrid optimization algorithm for depression detection using MRI and speech signal', *Biomedical Signal Processing and Control*, vol. 86, p. 105046, Sep. 2023, doi: 10.1016/j.bspc.2023.105046.
[39] W. Zhang, K. Mao, and J. Chen, 'A Multimodal Approach for Detection and Assessment of Depression Using Text, Audio and Video', *Phenomics*, May 2024, doi: 10.1007/s43657-023-00152-8.
[40] P. Singh, P. Raj, and V. P. Namboodiri, 'EDS pooling layer', *Image and Vision Computing*, vol. 98, p. 103923, Jun. 2020, doi: 10.1016/j.imavis.2020.103923.
[41] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, 'OpenFace 2.0: Facial Behavior Analysis Toolkit', in : *Proc. of the 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018)*, Xi'an, China: IEEE Explore, 2018, pp. 59–66. doi: doi: 10.1109/FG.2018.00019.
[42] B. Arroll *et al.*, 'Validation of PHQ-2 and PHQ-9 to Screen for Major Depression in the Primary Care Population', *The Annals of Family Medicine*, vol. 8, no. 4, pp. 348–353, Jul. 2010, doi: 10.1370/afm.1139.
[43] S. Sugiyanto, I. K. E. Purnama, E. M. Yuniarno, W. Anggraeni, and M. H. Purnomo, 'Depression Classification Based on Facial Action Unit Intensity Features Using CNN-Poolingless Framework', *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 5, pp. 172–187, 2024, doi: 10.22266/ijies2024.1031.15.