DETECTING DEPRESSION IN SOCIAL MEDIA USING MACHINE LEARNING

Ruoxi Ding[1] and Yu Sun[2]

[1]Woodbridge High School, 2 Meadowbrook, Irvine, CA 92604
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Social Media Depression Detection is an Intelligent System to automate the detection of Youth Depression with social media (Instagram) using AI and Deep Learning. The student is the targeted group because most students with depression express themselves on social media rather than seeking help from doctors. This app gathers captions and images from the user's personal Instagram profile through web scraping using Instagram private API to check whether or not the posts are depressive. The google cloud dataset supports the captions and pictures analysis performed by the app [6]. Caption sentiment analysis depends on sentiment analysis, and the pictures analysis depends on classifying images by custom labels. The app reports the image and the caption analysis results back to the user. Python is used for the back-end functionality, while Dart and Flutter are used for the front-end. It was tested by 2 experiments, the first experiments returned the feedback of 15 students demonstrates that the program has the capability of detecting depression through the captions with relatively high accuracy. The second experiment of testing the app functionality on the same account demonstrates that the program is stable and consistent. The purpose of the app is to detect depression at an early stage to prevent the condition from worsening.

KEYWORDS

NLP, Web Scraping, Machine Learning.

1. INTRODUCTION

Depression is one of the most frequent and devastating mental illnesses that has a significant social impact. It is common knowledge that mental health is an important aspect of public health. Depression is one of the most common causes of disability in the world. Millions of individuals suffer from depression around the world, according to the results of World Health Surveys (Moussavi et al., 2007) [1]. Recent research has attempted to use social media to detect depression since the patterns of ideas and thoughts represented in posted text and photos can, to some extent, mirror users' mental states [2]. According to WHO estimates from 2014, approximately 20% of children and adolescents have suffered mental illness, with half of these diseases beginning before the age of 14. Furthermore, mental and substance use disorders were responsible for almost 23% of all deaths worldwide. Because of the close interaction that exists between social media platforms and its users, these platforms have come to mirror the users' personal lives on a number of levels [3]. Apart from the severity of mental problems and their impact on one's mental and physical health, societal stigma (e.g., "mental disorders cannot be treated") and discrimination have caused individuals to be ignored by the community and to avoid receiving necessary treatment [3]. Mental illnesses, such as depression, should be

David C. Wyld et al. (Eds): ICAIT, CBIoT, WiMo, CRYPIS, ICDIPV, CAIML, NLCA - 2022
pp. 277-291, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121223

recognized early on to avoid the condition from worsening and to maximize the chances of patients receiving appropriate treatment.

Some techniques and systems have been proposed to detect depression in social media by asking patients to fill out interactive questionnaires that contain specific questions designed to detect depression, allowing the user to report whether they have depressive symptoms such as sadness or depressed mood, loss of enjoyment, major weight change, insomnia or excessive sleep, and others [4]. These proposals, however, presume that patients with depression are willing to reply to the survey, which is rarely the case in practice because most patients do not want to open their minds to interact with the survey, and some patients do not acknowledge having depression. Another method is to check for physical indicators of depression, such as back discomfort, headaches, limb pain, joint pain, gastrointestinal issues, sleep issues, and so on [4]. About half of those who suffer from depression are never recognized or treated, and not receiving treatment can be fatal [4]. Because doctors are required to physically contact patients in order to assist them find out the root of their symptoms, and because the majority of depression sufferers do not seek care, this method could not be too complex, and it frequently leads to patients' late depression. Patients can also write down their concerns about any specific depressive symptoms or unusual behaviors they are experiencing, as well as their detailed family history from relatives, but they will need medical assistance. This makes it difficult for patients who primarily use social media and refuse to seek medical assistance. The majority of research into how to direct mental illness in social media platforms has centered on feature engineering [3]. Researchers had to identify each mental disease by extracting traits that overlapped. This procedure isn't always accurate, and it's difficult to forecast which mental disease a patient will be diagnosed with.

In this paper, I follow the same line of research by finding the depressive symptoms and implementing it into the project. My goal is to detect depression in Instagram by analyzing the users' posts including texts, videos, and images [7]. My method is inspired by feature engineering, which is the process of transforming raw data into features that better represent the underlying problem to the predictive models [8]. The mobile app I created is an Intelligent System to automate the detection of Youth Depression with social media (Instagram) using AI and Deep Learning. I choose the student to be the targeted group because most students with depression are not likely to seek help and also teenagers use social media a lot. This app gathers captions, images, and videos from the user's personal Instagram profile to check whether or not the posts are depressive. The program utilized sentiment analysis in the text analysis part, google.cloud vision image annotator, image label filter, safe search, and face detection for the image analysis. The results splits into slightly depressive, depressive, and positive. The app also informs the user of the results in order to encourage them to get proper treatments and prevent possibilities of mental health-related problems.

I did two experiments to prove my program. The first experiment is conducted on 15 students who frequently make instagram posts. The feedback received demonstrates that 71% of slightly depressive students, 75% of depressive students, and 100% of positive students were correctly-identified for the caption analysis. For the image analysis, 70% of depressive students and 80% of positive students were correctly-identified. Considering the difficulty and the complexity of finding possible depressive symptoms only through the social media posts, my program that combines the captions and the image analysis proves that it can accurately detect depression in most of the cases. The second experiment was conducted based on the performance of the program 50 times for each of the 2 Instagram accounts: one with completely positive posts, and the other with completely depressive posts. The results received demonstrate that the caption and the image analysis were both 100% consistent. The average time taken for getting the final analysis results of the positive account is slightly greater than the depressive accounts since the positive account doubles the amount of posts of the depressive account. The 100% consistency of the 50 trials for each account proves that the analysis results generated by the program are trustworthy.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the tracking system, a few challenges have been identified as follows.

2.1. Collecting Data and Getting the Updated Instagram Posts

Collecting the data required for building the mobile program and getting the updated user posts on instagram are challenging [9]. It’s easy to get lots of users’ posts from a large dataset, but it doesn’t work for the purpose of gathering captions and videos from the updated version of the user's personal Instagram program. It’s complicated to scrape the data stored in another social media and implement my program to analyze these data. People want to collect information that ranges anywhere from collecting usernames, finding followers, collecting comments, and analyzing conversations that include the keywords. Also, sensitive data that can be used to manipulate others should not be gathered and used for any purposes. In addition, the data extracted from the Instagram social media platform is challenging due to the unstructured nature of the text posted by users [11]. The posts are introduced with misspelled words, new terms, character limitations, and syntactic errors when composing a message.

2.2. Image Analysis

Since Instagram posts are all published with images, images can better express feelings than words sometimes through the color, darkness, character, and feeling it creates [12]. In order to build the depression detection app, image analysis is required to determine whether the user is positive, slightly depressive, or depressive. Conducting the image analysis is complicated because there’s no standard. For example, dark color doesn’t mean that the picture is sad and depressing, and bright color also doesn’t mean that the picture is positive and joyful. It might be easy for humans to look over the pictures and decide which category that certain picture should be in, but it’s complicated for machines to analyze. In addition, pictures also include different elements or symbols that represent a certain thing, which makes it complex for computers. Irony has also been increasingly used in images, which often trick computers to consider the image by completely off or opposite meaning.

2.3. Dart Front-end Development

Since the Instagram Depression Detection is an app that I’m building, I need to develop the front-end, or the client side to make the app user-friendly. I also want to publish the app on both IOS and Android systems so I choose Dart, which is a language that I don’t have any experience with [13]. I encountered a lot of trouble during the process of switching pages and building the necessary elements, such as enabling the user to enter username and password, linking the analysis result with the result page, demonstrate the users’ pictures and captions along with the analysis, and changing the settings including fonts, backgrounds, size, page size. I also need to take into consideration different mobile phones' sizes.

3. SOLUTION

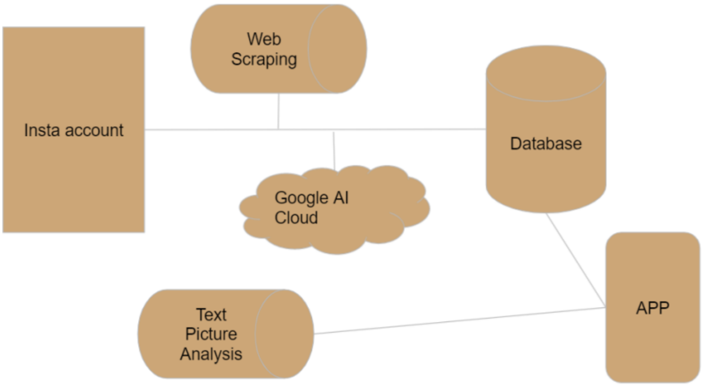
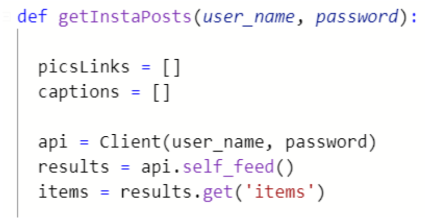


Figure 1. Flow of The Process

DepressionEyes is an Intelligent System to automate the detection of Youth Depression with social media (Instagram) using AI and Deep Learning. As shown in figure 1, DepressionEyes gathers captions and images from the user's personal Instagram profile through web scraping with the help of client in the Instagram private API [6]. Google cloud generates a large database of text sentiment analysis and pictures labels [10]. The captions and pictures analysis performed by the app are supported by google cloud dataset. Caption analysis is dependent on google cloud language_v1 package text sentiment analysis. All text input will receive a score from -1 to 1, -1 being the extreme negative, and 1 being the extreme positive. The pictures analysis is dependent on google cloud vision image label filter, label detection, safe search annotation, and human face annotations. Due to the large dataset contained in google cloud vision, image input will be detected either as depressive or not depressive. The app divided the total number of the images received from the user's Instagram posts by the total number of depressive images detected to find and report whether half of the user's pictures are depressive depending on the percentage of depressive images among all. For the caption analysis, the app finds the average of the sentiment analysis performed on all captions, and splits the results into slightly depressive, depressive, and positive captions. In the front-end app building, I wrote the code in dart and flutter, which I included five pages: about page, info page, login page, main page, and pictures page. The app links the back-end python code with the front-end in the info page, so the results will be demonstrated after the user puts in their instagram username and password.

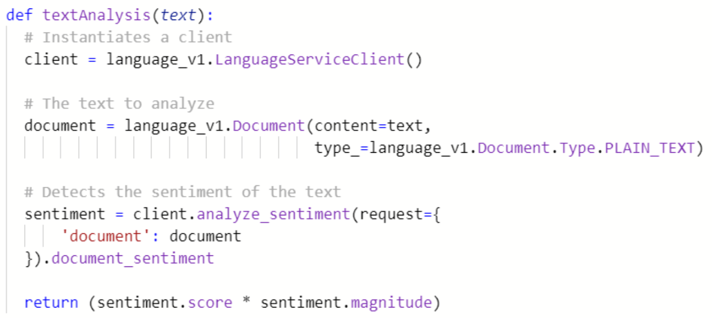
This part explains how I implemented the Back-end of DepressionEyes written in Python. As shown in figure 2, the instagram private api client class enables the user to pass in their username and password and the program then has the ability to access the user's instagram. The program used the self_feed() function to get authenticated user's own feed [6]. Since the program only needs captions and pictures, it collects information inside 'item' among the user's own feed. The pictures are presented as links, which are stored in the picsLinks list. The captions are stored in the captions list.



```
def getInstaPosts(user_name, password):

    picsLinks = []
    captions = []

    api = Client(user_name, password)
    results = api.self_feed()
    items = results.get('items')
```

Figure 2. Getting User’s Posts

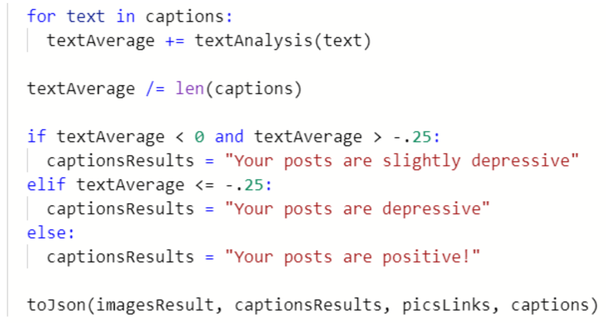
In order to determine whether the input captions are depressive, the program included a textAnalysis function defined with a parameter text, which received the passed-in captions. The text sentiment analysis is dependent on the google cloud language_v1 package [7]. The program first instantiates a client through the LanguageServiceClient() function, then passes the caption input to the Document() function. The program detects the sentiment of the captions using the analyze_sentiment() function and the document_sentiment. document_sentiment contains the overall sentiment of the document, which consists of score and magnitude. The sentiment score runs from -1.0 (negative) to 1.0 (positive) and refers to the text's total emotional learning. The document's magnitude reveals how much emotional content it contains. Magnitude is significant when analyzing documents with a neutral score since it allows one to determine how much emotional content is present. The textAnalysis(text) function returns the sentiment score multiplied by the magnitude as a result.



```
def textAnalysis(text):
    # Instantiates a client
    client = language_v1.LanguageServiceClient()

    # The text to analyze
    document = language_v1.Document(content=text,
                                    type_=language_v1.Document.Type.PLAIN_TEXT)

    # Detects the sentiment of the text
    sentiment = client.analyze_sentiment(request={
        'document': document
    }).document_sentiment

    return (sentiment.score * sentiment.magnitude)
```

Figure 3. Text Analysis Function

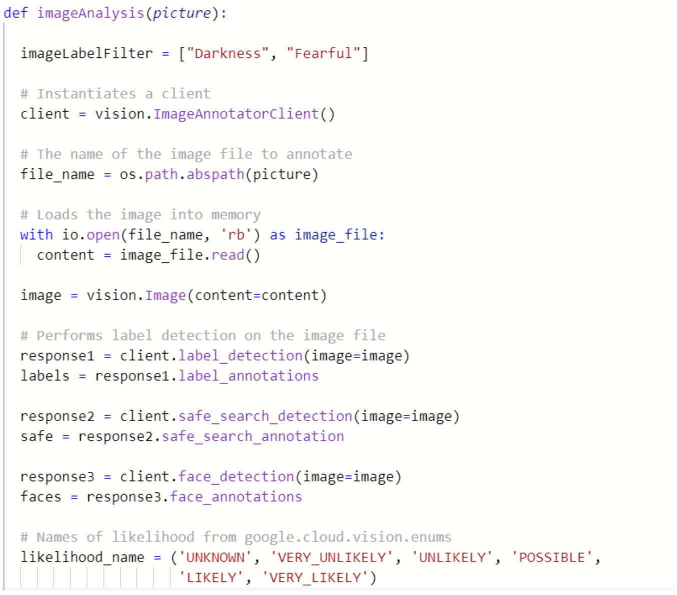
The program finds the scores of the captions stored in the captions list. As shown in figure 4, the captions average is either identified as slightly depressive, depressive, and positive. Average score above 0.0 is considered positive, average below -0.25 is considered negative, and average between -0.25 and 0.0 is considered slightly depressive.



```
for text in captions:
    textAverage += textAnalysis(text)

textAverage /= len(captions)

if textAverage < 0 and textAverage > -.25:
    captionsResults = "Your posts are slightly depressive"
elif textAverage <= -.25:
    captionsResults = "Your posts are depressive"
else:
    captionsResults = "Your posts are positive!"

toJson(imagesResult, captionsResults, picsLinks, captions)
```

Figure 4. Caption Results Supported by The Text Analysis Function

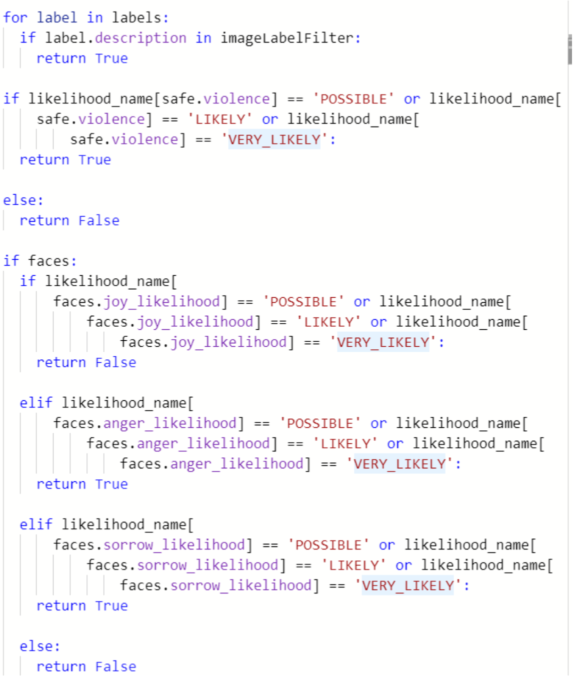
DepressionEyes utilized google cloud vision AI to conduct the images analysis [8]. With AutoML Vision, Vision AI employs machine learning to analyze images with industry-leading prediction accuracy, train machine learning models that classify images by custom labels, and detect objects and faces, read handwriting, and construct important image metadata [8]. After instantiating a client and loading the image into memory, the program performs label detection on the image file. It stored the labels, safe search results, and the face features results of the images in different variables as shown in figure 5.



```
def imageAnalysis(picture):

    imageLabelFilter = ["Darkness", "Fearful"]

    # Instantiates a client
    client = vision.ImageAnnotatorClient()

    # The name of the image file to annotate
    file_name = os.path.abspath(picture)

    # Loads the image into memory
    with io.open(file_name, 'rb') as image_file:
        content = image_file.read()

    image = vision.Image(content=content)

    # Performs label detection on the image file
    response1 = client.label_detection(image=image)
    labels = response1.label_annotations

    response2 = client.safe_search_detection(image=image)
    safe = response2.safe_search_annotation

    response3 = client.face_detection(image=image)
    faces = response3.face_annotations

    # Names of likelihood from google.cloud.vision.enums
    likelihood_name = ('UNKNOWN', 'VERY_UNLIKELY', 'UNLIKELY', 'POSSIBLE',
                       'LIKELY', 'VERY_LIKELY')
```

Figure 5. Label Detection in Image Analysis

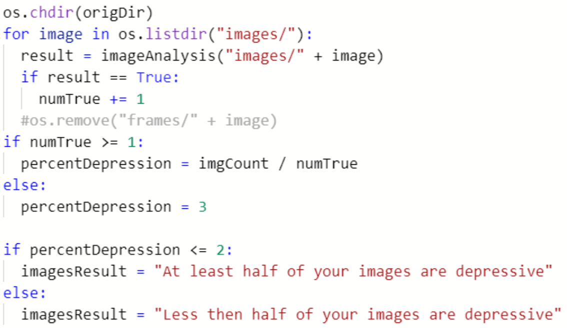
If the labels in the image include “darkness” and “fearful” defined in figure 5, the picture is considered as depressive. If the safe violence presented in the picture, it’s considered depressive. Considering the face features in case some posts included faces, faces with anger and sorrow are considered depressive, and faces with joy are considered positive as shown in figure 6.



```
for label in labels:
  if label.description in imageLabelFilter:
    return True

if likelihood_name[safe.violence] == 'POSSIBLE' or likelihood_name[
    safe.violence] == 'LIKELY' or likelihood_name[
        safe.violence] == 'VERY_LIKELY':
  return True

else:
  return False

if faces:
  if likelihood_name[
      faces.joy_likelihood] == 'POSSIBLE' or likelihood_name[
          faces.joy_likelihood] == 'LIKELY' or likelihood_name[
              faces.joy_likelihood] == 'VERY_LIKELY':
    return False

  elif likelihood_name[
      faces.anger_likelihood] == 'POSSIBLE' or likelihood_name[
          faces.anger_likelihood] == 'LIKELY' or likelihood_name[
              faces.anger_likelihood] == 'VERY_LIKELY':
    return True

  elif likelihood_name[
      faces.sorrow_likelihood] == 'POSSIBLE' or likelihood_name[
          faces.sorrow_likelihood] == 'LIKELY' or likelihood_name[
              faces.sorrow_likelihood] == 'VERY_LIKELY':
    return True

  else:
    return False
```

Figure 6. Categorizing Different Images

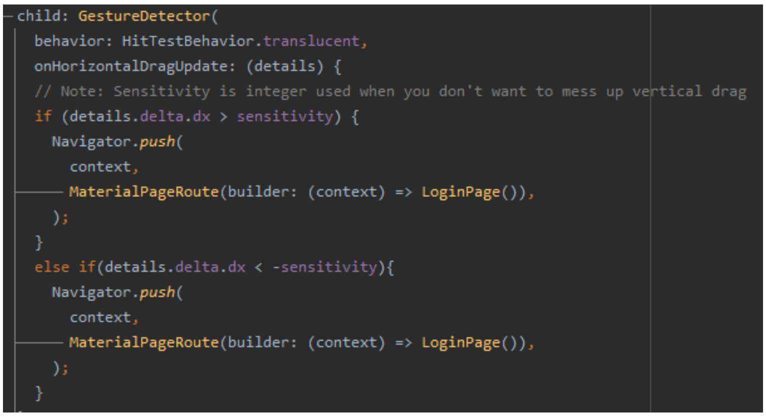
In figure 7, the program finds the depression percentages by dividing the total images with the depressive images. It reports whether or not half of the images are depressive.



```
os.chdir(origDir)
for image in os.listdir("images/"):
  result = imageAnalysis("images/" + image)
  if result == True:
    numTrue += 1
  #os.remove("frames/" + image)
if numTrue >= 1:
  percentDepression = imgCount / numTrue
else:
  percentDepression = 3

if percentDepression <= 2:
  imagesResult = "At least half of your images are depressive"
else:
  imagesResult = "Less then half of your images are depressive"
```

Figure 7. Images Results Supported by The Images Analysis Function

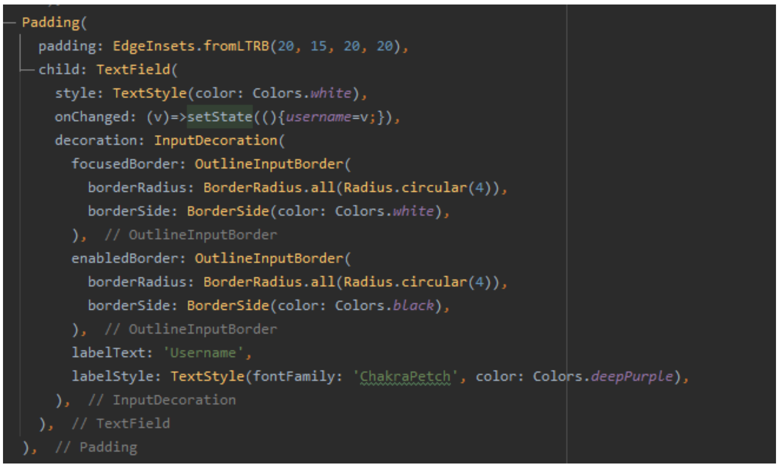
I chose flutter, an open-source UI software development kit created by Google, for the front-end development of my program because flutter can be used to develop cross platform applications for iOS, Android, Linux, Mac, Windox, etc. There are five pages in the programs, the main page that leads the users to the program, the login page that enables the users to put in their Instagram username and password, about page that give instructions on how the program works to the users, info page that demonstrates the caption and image analysis of the users' instagram results, and pictures page that demonstrates all pictures in users' instagram posts. Considering switching between screens, I used a gesture detector to navigate to another page through the sensitivity of the hit test behavior as shown in figure 8.



```
child: GestureDetector(
  behavior: HitTestBehavior.translucent,
  onHorizontalDragUpdate: (details) {
  // Note: Sensitivity is integer used when you don't want to mess up vertical drag
  if (details.delta.dx > sensitivity) {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => LoginPage()),
    );
  }
  else if(details.delta.dx < -sensitivity){
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => LoginPage()),
    );
  }
```

Figure 8. Swipe Picture from Main Page to Login Page

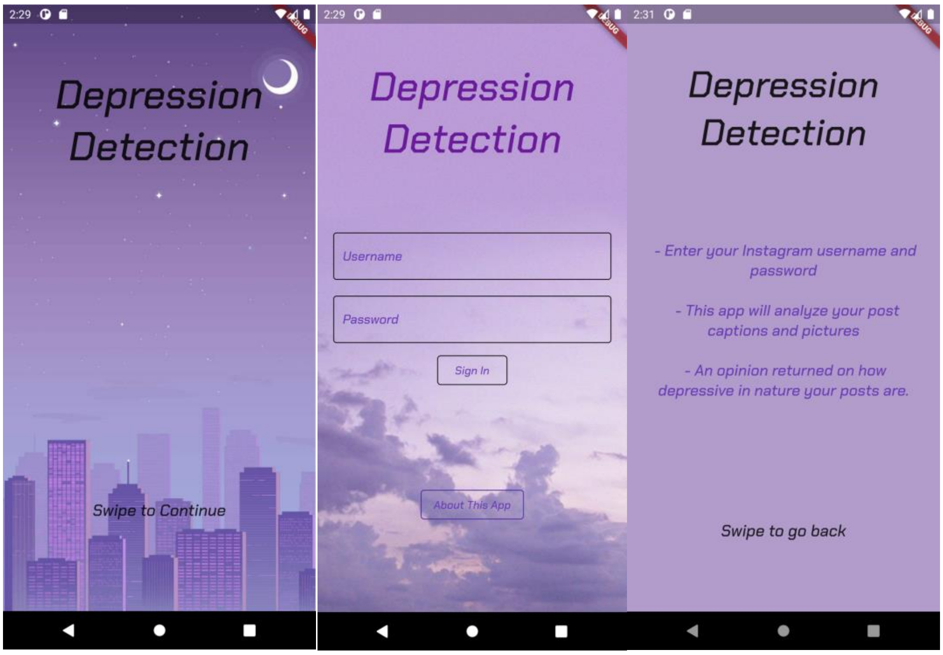
On the login page, I used the input decoration, the outline input border, and the ChakraPetch fonts inside the input box as shown in figure 9. The first box on the page is asking for the username, and the box below is asking for the password. A button to the about page is also on the login page.



```
Padding(
  padding: EdgeInsets.fromLTRB(20, 15, 20, 20),
  child: TextField(
    style: TextStyle(color: Colors.white),
    onChanged: (v)=>setState((){username=v;}),
    decoration: InputDecoration(
      focusedBorder: OutlineInputBorder(
        borderRadius: BorderRadius.all(Radius.circular(4)),
        borderSide: BorderSide(color: Colors.white),
      ),  // OutlineInputBorder
      enabledBorder: OutlineInputBorder(
        borderRadius: BorderRadius.all(Radius.circular(4)),
        borderSide: BorderSide(color: Colors.black),
      ),  // OutlineInputBorder
      labelText: 'Username',
      labelStyle: TextStyle(fontFamily: 'ChakraPetch', color: Colors.deepPurple),
    ),  // InputDecoration
  ),  // TextField
),  // Padding
```

Figure 9. Login Page Input Box

Figure 10 demonstrates the first three screens of the application. The first one from the left is the main page, the second one from the left is the login page, and the third one from the left is the about page. The theme is purple, which remained consistent throughout the application.



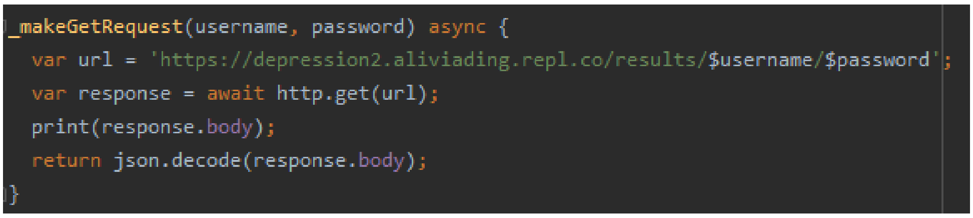
- Enter your Instagram username and password

- This app will analyze your post captions and pictures

- An opinion returned on how depressive in nature your posts are.

Figure 10. Demonstration of Main Page, Login Page, and About Page

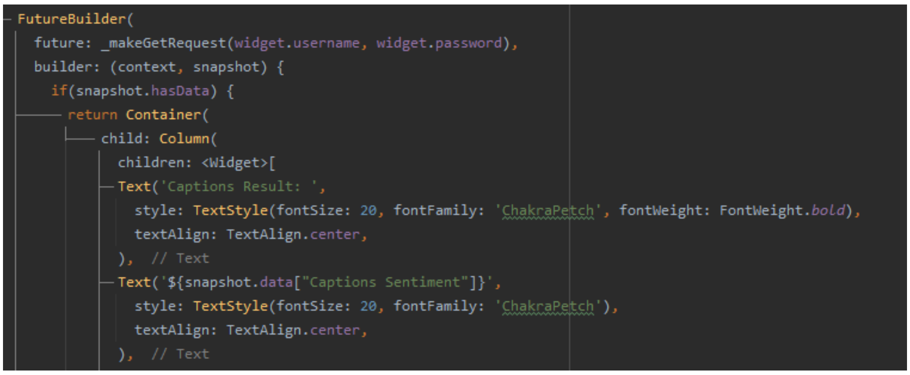
As shown in figure 11, I linked my back-end Python program with flutter in the Info Page by making a get request to my program on repl.it with the username and the password passed.



```
_makeGetRequest(username, password) async {
  var url = 'https://depression2.aliviading.repl.co/results/$username/$password';
  var response = await http.get(url);
  print(response.body);
  return json.decode(response.body);
}
```

Figure 11. Info Page Linking to Back-end

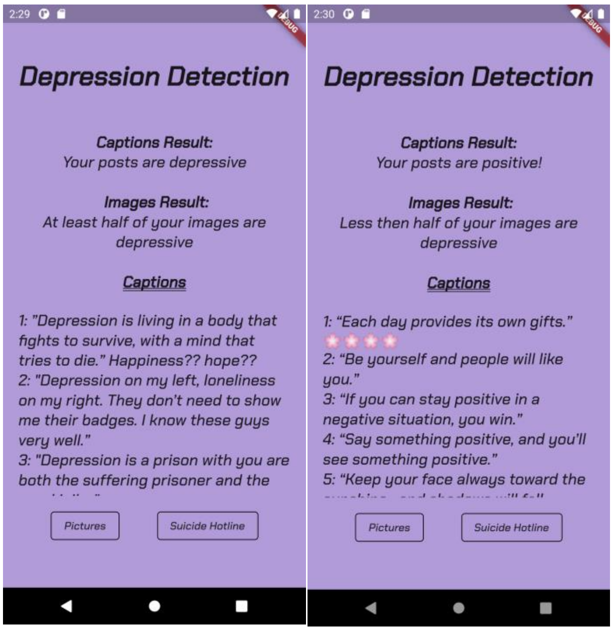
The info page also serves the purpose of demonstrating the caption and the picture analysis results. As shown in figure 12, I received the caption results from the “Captions Sentiment” element of the analysis, and demonstrated the results in ChakraPetch 20 fonts. The pictures results demonstrate in similar ways.



```
FutureBuilder(
  future: _makeGetRequest(widget.username, widget.password),
  builder: (context, snapshot) {
    if(snapshot.hasData) {
      return Container(
        child: Column(
          children: <Widget>[
          Text('Captions Result: ',
            style: TextStyle(fontSize: 20, fontFamily: 'ChakraPetch', fontWeight: FontWeight.bold),
            textAlign: TextAlign.center,
          ),  // Text
          Text('${snapshot.data["Captions Sentiment"]}',
            style: TextStyle(fontSize: 20, fontFamily: 'ChakraPetch'),
            textAlign: TextAlign.center,
          ),  // Text
```

Figure 12. Info Page Elements

Figure 13 is a demonstration of the info pages. The one on the left is tested by a depressive instagram account, so the caption and picture results all show that the Instagram account is depressive. The one on the right is tested by a positive instagram account, so the caption and picture results all show that this instagram account is positive.



Depression Detection

Captions Result:
Your posts are depressive

Images Result:
At least half of your images are depressive

Captions

1: "Depression is living in a body that fights to survive, with a mind that tries to die." Happiness?? hope??
2: "Depression on my left, loneliness on my right. They don't need to show me their badges. I know these guys very well."
3: "Depression is a prison with you are both the suffering prisoner and the

Depression Detection

Captions Result:
Your posts are positive!

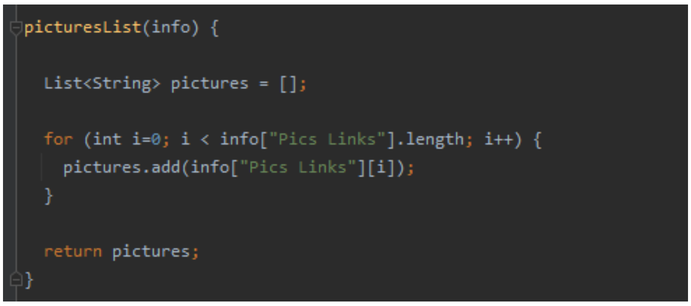
Images Result:
Less then half of your images are depressive

Captions

1: "Each day provides its own gifts." 🌸🌸🌸🌸
2: "Be yourself and people will like you."
3: "If you can stay positive in a negative situation, you win."
4: "Say something positive, and you'll see something positive."
5: "Keep your face always toward the

Figure 13. Depressive and Positive Info Page Demonstrations

As shown in figure 13, there's a picture button linking to the picture page that displays all pictures from the user posts. The program converted the pictures to links, and I stored and displayed the link in the pictures array as shown in figure 14. The program also have a "Suicide Hotline" button on the info page, which linked to https://suicidepreventionlifeline.org/ in case the users need help.



```
picturesList(info) {

  List<String> pictures = [];

  for (int i=0; i < info["Pics Links"].length; i++) {
    pictures.add(info["Pics Links"][i]);
  }

  return pictures;
}
```

Figure 14. Getting The Pictures

Figure 15 demonstrates the pictures page of the same positive and depressive instagram accounts the program used for the Info page demonstration. The one on the left has depressive Instagram posts and the one on the left has positive instagram posts. The picture pages listed all pictures in the posts.

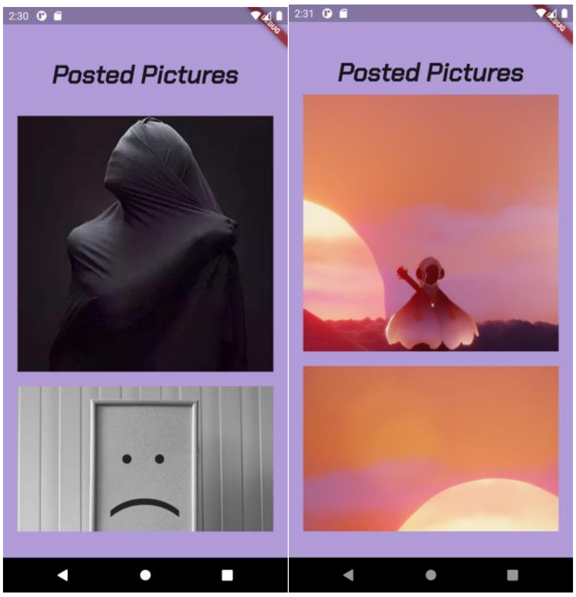


Figure 15. Depressive and Positive Picture Pages Demonstrations

4. EXPERIMENT

Experiment 1

The program served the purpose of detecting depression among the youth generation through their social media posts. In order to test the efficiency and the accuracy of the program, I conducted an experiment on 15 students that frequently used Instagram. They put in their username and password to the program and check if the captions and the image analysis presented match what they think. The purpose of the experiments is to find out the percentage of students that's detected currently.

Table 1 demonstrates the accuracy of the text sentiment analysis. Rows represented the number of students who considered themselves as either depressive, slightly depressive, or positive. Columns representing the number of students are being correctly-identified, wrongly identified, and calculated the match percentage. For each row, the correctly-identified and the wrongly-identified students add up to be the total number of students that considered themselves as the corresponding category.

Table1. Text Sentiment Analysis Results

	Correctly-identified	Wrongly-identified	Percentage match
Depressive	3	1	75%
Slightly Depressive	5	2	71%
Positive	4	0	100%

Table 2 demonstrates the accuracy of the image sentiment analysis. Rows represented the number of students who considered themselves as either depressive or positive. Columns representing the number of students that are being correctly-identified, wrongly identified, and the percentage match shows. The numbers of correctly and wrongly identified added up the total students of each corresponding category.

Table 2. Image Analysis Results

	Correctly-identified	Wrongly-identified	Percentage match
Depressive	7	3	70%
Positive	4	1	80%

Experiment 2

In order to test the consistency and also the accuracy of the program, the second experiment tested the program performance on two Instagram accounts, each for 50 trials. One of the accounts is intended to be depressive, with depressive captions (i.e., depressive quotes) and depressive images, and a total of five posts. The other account is designed to be positive that includes positive contents with a total of 10 posts. This experiment measured and recorded the caption and image analysis result along with the time(sec) takes for the analysis to process. The purpose of the experiments is to find out how accurate and consistent the program is.

The caption analysis and picture analysis results for all 50 trails for the positive account and the depressive account are the same, as shown in table 3. Therefore, the percentage match for both is 100 percent. The average time spent for getting a result is 16.5 seconds for the positive account, and 14.25 seconds for the depressive account. The depressive account has less posts than the positive accounts, and the time taken is more consistent than the positive account with more posts.

Table 3. Consistency Tests on Two Accounts with either Positive or Negative Posts

	consistency of caption analysis	consistency of image analysis	percentage match (accuracy for caption analysis)	percentage match (accuracy for image analysis)	Average time spent for result (caption analysis + image analysis)
positive	100% same	100% same	100%	100%	16.5 sec
depressive	100% same	100% same	100%	100%	14.25 sec

The first experiment demonstrates that 71% of slightly depressive students, 75% of depressive students, and 100% of positive students were correctly-identified for the caption analysis. The slightly depressive students have the lowest percentage because their posts are more neutral than students who are strongly depressive or positive. This proves that the program has the capability of detecting depression through the captions. It has the ability to improve through training the model by captions that are not extremely negative or positive. For the image analysis, 70% of depressive students and 80% of positive students were correctly-identified. This shows that the image analysis has lower accuracy than the caption analysis, but it can still detect most of the depressive pictures. Combining the captions analysis and the image analysis, the program proves that it can accurately detect depression in most of the cases.

The second experiment demonstrates that the caption and the image analysis were both 100% consistent. The average time spent for the analysis results including the caption and image analysis for the positive account is 16.5 seconds, and the average time for the depressive account is 14.25 seconds. Considering that the positive account doubles the amount of posts of the depressive account, longer time is reasonable. Also, the 100% consistency proves that this program is trustworthy.

5. RELATED WORK

Many well regarded scales and criteria have been developed based on the user research or questionnaire survey [5]. Beck's Depression Inventory [Beck et al., 1961], for example, consists of 21 questions about the mental and physiological status of the user. Another example is the CES-D Scale [Radloff, 1977], which includes 20 items about mental health issues such as users' guilt sentiments and sleep patterns. My program directly analyzes the posts in social media instead of asking the users’ survey questions. One of the strengths of my work is that detection can be made even when the patient refuses to answer the questions.

With the passage of time, we, including depressed users, are nearly unable to live without social media. As a result, researchers began to study the online behaviors of depressed individuals. Park et al. [2012] used real-time moods acquired from Twitter users to investigate the usage of language in portraying depressive moods as a preliminary study. Park et al. [2013] performed face-to-face interviews with 14 active Twitter users to investigate their depressive behaviors in their follow-up study. My program also used social media as a platform to detect depression. My program performed a text analysis on the user's caption. In addition to that, my program has image analysis which enables posts with only images to be tested. The benefit of their program is that they conducted interviews with active Twitter users to explore their depressive behaviors, so they know how people with depression act in social media.

Another study (Wang et al., 2013) presented a method for detecting depression in microblogs based on subject-dependent sentiment analysis. They think that a lack of good emotions and a lack of negative emotions are essential markers of depression. They used ten psychologically-based features of depressed users: the number of emoticons, interaction features (how users connect with one another), and behavior features (frequencies of the posting, active period). My program did similar things to detect depression, such as the sentiment analysis of the posts. One difference is that I only considered text/pictures depressive when it contains negative contents, Wang’s approach classifies microblog as depressive when there’s lack of positive emotions. This may help prohibit the omission of anyone with depression, but it might not be specific enough.

6. CONCLUSIONS

Social Media Depression Detection is an Intelligent System that uses AI and Deep Learning to automate the detection of Youth Depression on social media (Instagram). Because most students with depression express themselves on social media rather than seeking medical treatment, they are the target group. This software uses web scraping and the Instagram private API to collect captions and images from a user's personal Instagram profile in order to determine whether or not the postings are depressing. The descriptions and image analysis performed by the app are supported by the Google cloud dataset. Sentiment analysis is required for caption sentiment analysis, while image classification by custom labels is required for image analysis. The image and caption analysis findings are returned to the user via the app. The backend is written in Python, while the front-end is written in Dart and Flutter [14]. It was put to the test in two ways. The first trial yielded input from 15 students, demonstrating that the algorithm is capable of identifying depression through captions with a high degree of accuracy. The app's operation was tested again on the same account, demonstrating that the software is stable and consistent.

For the image analysis, 70% of depressive students and 80% of positive students were correctly-identified. This shows that the image analysis has lower accuracy than the caption analysis. The pictures that the user posts in Instagram is important to the result, so the program needs to improve the image analysis accuracy. Also, this program can only be conducted on the user’s own Instagram, because it requires username and password. It would be a lot better if it can be used to test other public accounts. An extension that contains these functionality and can be used to test all public accounts is more ideal. The look of the pages can also be more attractive.

In order to improve the accuracy of the image analysis, I will try more libraries or use the library to scratch my own algorithm [15]. For example, I will try the openCV package with python, tensorflow with vuforia, and enhance the way it classifies something as depressive. I will make the mobile app function the same on the computer and potentially make it an extension.

REFERENCES

[1] Stankevich, Maxim, et al. "Feature Engineering for Depression Detection in Social Media." ICPRAM. 2018.

[2] Lin, Chenhao, et al. "Sensemood: Depression detection on social media." Proceedings of the 2020 International Conference on Multimedia Retrieval. 2020.

[3] Orabi, Ahmed Husseini, et al. "Deep learning for depression detection of twitter users." Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic. 2018.

[4] Beck, Aaron T. "The diagnosis and management of depression." (1973).

[5] Shen, Guangyao, et al. "Depression Detection via Harvesting Social Media: A Multimodal Dictionary Learning Solution." IJCAI. 2017.

[6] Garraghan, Peter, Paul Townend, and Jie Xu. "An analysis of the server characteristics and resource utilization in google cloud." 2013 IEEE International Conference on Cloud Engineering (IC2E). IEEE, 2013.

[7] Instagram, Instagram. "Instagram." Facebook, https://www. instagram. com (2016).

[8] Turner, C. Reid, et al. "A conceptual basis for feature engineering." Journal of Systems and Software 49.1 (1999): 3-15.

[9] Coombs, Clyde H. "A theory of data." (1964).

[10] Data, Modeling Historical, and Guide User. Database Design. Perancangan Basis Data) merupakan salah satu, 2015.

[11] Bogdanov, Evgeny, et al. "A social media platform in higher education." Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON). IEEE, 2012.

[12] Koenderink, Jan J. "The structure of images." Biological cybernetics 50.5 (1984): 363-370.

[13] Al-Qershi, Fattoh, et al. "Android vs. iOS: The security battle." 2014 World Congress on Computer Applications and Information Systems (WCCAIS). IEEE, 2014.

[14] De Smedt, Tom, and Walter Daelemans. "Pattern for python." The Journal of Machine Learning Research 13.1 (2012): 2063-2067.

[15] Freese, Frank. "Testing accuracy." Forest Science 6.2 (1960): 139-45.