Decoding Emotions Through Sentiment Analysis of Social Media Conversations

Requirements:
Install the necessary libraries:
pip install tweepy textblob nltk pandas matplotlib seaborn wordcloud

Python Source Code:

```
import tweepy
import pandas as pd
from textblob import TextBlob
import nltk
from nltk.corpus import stopwords
import re
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import seaborn as sns

nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

# Twitter API credentials
api_key = 'YOUR_API_KEY'
api_secret = 'YOUR_API_SECRET'
access_token = 'YOUR_ACCESS_TOKEN'
access_token_secret = 'YOUR_ACCESS_TOKEN_SECRET'

# Authenticate with Twitter
auth = tweepy.OAuthHandler(api_key, api_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# Fetch tweets
def fetch_tweets(keyword, count=100):
    tweets = tweepy.Cursor(api.search_tweets, q=keyword, lang='en').items(count)
      tweets_data = [{'text': tweet.text, 'created_at': tweet.created_at} for tweet in
tweets]
    return pd.DataFrame(tweets_data)

# Clean text
def clean_text(text):
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
    text = re.sub(r'\@\w+|\#', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    text = text.lower()
    text = ' '.join(word for word in text.split() if word not in stop_words)
    return text

# Sentiment and emotion detection
def analyze_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    if polarity > 0.2:
```

```
        emotion = 'Happy'
    elif polarity < -0.2:
        emotion = 'Angry'
    elif polarity == 0:
        emotion = 'Neutral'
    else:
        emotion = 'Sad'
    return polarity, emotion

# Main analysis
def main():
    keyword = input("Enter a keyword or hashtag: ")
    df = fetch_tweets(keyword)
    df['clean_text'] = df['text'].apply(clean_text)
                df[['polarity',    'emotion']]    =    df['clean_text'].apply(lambda    x:
pd.Series(analyze_sentiment(x)))

    # Display sample
    print(df[['text', 'emotion']].head())

    # Visualization
    plt.figure(figsize=(10,6))
    sns.countplot(data=df, x='emotion', palette='viridis')
    plt.title('Emotion Distribution')
    plt.xlabel('Emotion')
    plt.ylabel('Count')
    plt.show()

    # Word Cloud
    all_words = ' '.join(df['clean_text'])
                            wordcloud       =       WordCloud(width=800,       height=400,
background_color='white').generate(all_words)
    plt.figure(figsize=(10, 6))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title('Word Cloud of Tweets')
    plt.show()

if __name__ == "__main__":
    main()
```