Sign Language to Audio using Computer Vision (LSTM Model)

In [1]:
```
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import mediapipe as mp
```

Keypoints using MP Holistic

In [2]:
```
mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities
```

In [4]:
```
def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB
    image.flags.writeable = False                  # Image is no longer writeable
    results = model.process(image)                 # Make prediction
    image.flags.writeable = True                   # Image is now writeable
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR COVERSION RGB 2 BGR
    return image, results
```

In [5]:
```
def draw_landmarks(image, results):
    mp_drawing.draw_landmarks(image, results.face_landmarks, mp_holistic.FACEMESH_C
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNE
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND
```

In [6]:
```
def draw_styled_landmarks(image, results):
    # Draw face connections
    mp_drawing.draw_landmarks(image, results.face_landmarks, mp_holistic.FACEMESH_C
                              mp_drawing.DrawingSpec(color=(80,110,10), thickness=1,
                              mp_drawing.DrawingSpec(color=(80,256,121), thickness=1
                              )
    # Draw pose connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNE
                              mp_drawing.DrawingSpec(color=(80,22,10), thickness=2,
                              mp_drawing.DrawingSpec(color=(80,44,121), thickness=2,
                              )
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_
                              mp_drawing.DrawingSpec(color=(121,22,76), thickness=2,
                              mp_drawing.DrawingSpec(color=(121,44,250), thickness=2
                              )
    # Draw right hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND
                              mp_drawing.DrawingSpec(color=(245,117,66), thickness=2
```

```
                                mp_drawing.DrawingSpec(color=(245,66,230), thickness=2
                                )
```

In [12]:

```
cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5
    while cap.isOpened():

        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)
        print(results)

        if results.left_hand_landmarks is not None:
            # Get the length of left hand landmarks
            num_left_hand_landmarks = len(results.left_hand_landmarks.landmark)
            print(f"Number of landmarks detected on the left hand: {num_left_hand_l
        else:
            print("No landmarks detected on the left hand.")
        # Draw landmarks
        draw_styled_landmarks(image, results)

        # Show to screen
        cv2.imshow('OpenCV Feed', image)

        # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

```
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
<class 'mediapipe.python.solution_base.SolutionOutputs'>
Number of landmarks detected on the left hand: 21
```

In [13]:

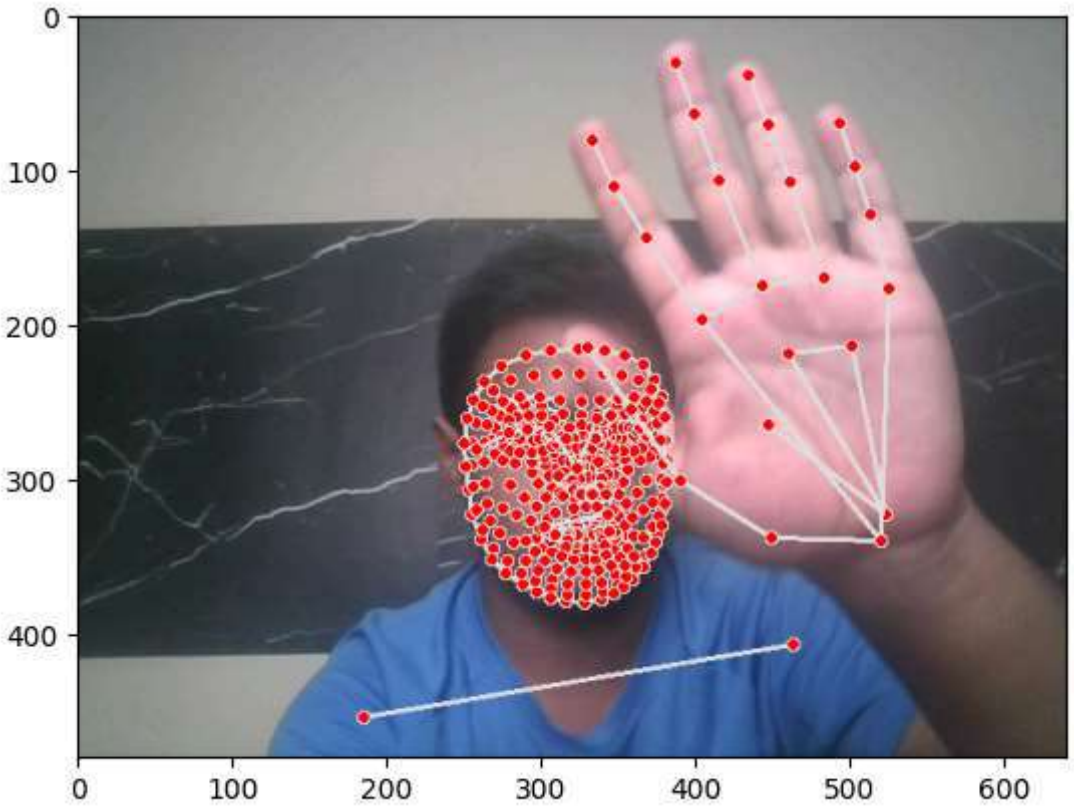
```
draw_landmarks(frame, results)
```

In [14]:

```
plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
```

Out[14]:

```
<matplotlib.image.AxesImage at 0x210685a65c0>
```



Extract Keypoint Values

In [15]:
```
len(results.left_hand_landmarks.landmark)
```

Out[15]: 21

In [16]:
```
pose = []
for res in results.pose_landmarks.landmark:
    test = np.array([res.x, res.y, res.z, res.visibility])
    pose.append(test)
```

In [17]:
```
pose = np.array([[res.x, res.y, res.z, res.visibility] for res in results.pose_land
face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark])
lh = np.array([[res.x, res.y, res.z] for res in results.left_hand_landmarks.landmar
rh = np.array([[res.x, res.y, res.z] for res in results.right_hand_landmarks.landma
```

In [18]:
```
face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark])
    if results.face_landmarks else np.zeros(1404)
```

In [7]:
```
def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in results.pose_
    face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landma
    lh = np.array([[res.x, res.y, res.z] for res in results.left_hand_landmarks.lan
    rh = np.array([[res.x, res.y, res.z] for res in results.right_hand_landmarks.la
    return np.concatenate([pose, face, lh, rh])
```

In [ ]:
```
result_test = extract_keypoints(results)
```

In [ ]:
```
result_test
```

In [22]:
```
np.save('0', result_test)
```

In [23]:
```
np.load('0.npy')
```

Out[23]:
```
array([ 0.50568753,  0.61139166, -1.01560879, ...,  0.        ,
        0.        ,  0.        ])
```

Setup Folders for Collection

In [24]:
```
# Path for exported data, numpy arrays

DATA_PATH = os.path.join(r'D:\data science\Practice\asl\MP_data')
os.makedirs(DATA_PATH, exist_ok=True)

# Actions that we try to detect
actions = np.array(['hello', 'thanks', 'iloveyou'])

# Thirty videos worth of data
no_sequences = 30

# Videos are going to be 30 frames in length
sequence_length = 30

# Folder start
start_folder = 30
```

In [25]:
```
for action in actions:
    for sequence in range(no_sequences):
        try:
            os.makedirs(os.path.join(DATA_PATH, action, str(sequence)))
        except:
            pass
```

Collect Keypoint Values for Training and Testing

In [26]:
```
cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5

    # NEW LOOP
    # Loop through actions
    for action in actions:
        # Loop through sequences aka videos
        for sequence in range(no_sequences):
            # Loop through video length aka sequence length
            for frame_num in range(sequence_length):

                # Read feed
                ret, frame = cap.read()
```

```
                # Make detections
                image, results = mediapipe_detection(frame, holistic)
#                 print(results)

                # Draw landmarks
                draw_styled_landmarks(image, results)

                # NEW Apply wait logic
                if frame_num == 0:
                    cv2.putText(image, 'STARTING COLLECTION', (120,200),
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4, cv2.LINE
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.f
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.L
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)
                    cv2.waitKey(2000)
                else:
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.f
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.L
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)

                # NEW Export keypoints
                keypoints = extract_keypoints(results)
                npy_path = os.path.join(DATA_PATH, action, str(sequence), str(frame
                np.save(npy_path, keypoints)

                # Break gracefully
                if cv2.waitKey(10) & 0xFF == ord('q'):
                    break

    cap.release()
    cv2.destroyAllWindows()
```

In [158…
```
cap.release()
cv2.destroyAllWindows()
```

Preprocess Data and Create Labels and Features

In [27]:
```
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
```

In [28]:
```
label_map = {label:num for num, label in enumerate(actions)}
```

In [29]:
```
label_map
```

Out[29]:
```
{'hello': 0, 'thanks': 1, 'iloveyou': 2}
```

In [31]:
```
sequences, labels = [], []
for action in actions:
    for sequence in range(no_sequences):
        window = []
```

```
        for frame_num in range(sequence_length):
            res = np.load(os.path.join(DATA_PATH, action, str(sequence), "{}.npy".f
            window.append(res)
        sequences.append(window)
        labels.append(label_map[action])
```

In [32]:
```
np.array(sequences).shape
```

Out[32]: (90, 30, 1662)

In [33]:
```
np.array(labels).shape
```

Out[33]: (90,)

In [34]:
```
X = np.array(sequences)
```

In [35]:
```
X.shape
```

Out[35]: (90, 30, 1662)

In [36]:
```
y = to_categorical(labels).astype(int)
```

In [37]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05)
```

In [38]:
```
y_test.shape
```

Out[38]: (5, 3)

Build and Train LSTM Neural Network

In [39]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.callbacks import TensorBoard
```

In [40]:
```
log_dir = os.path.join('Logs')
tb_callback = TensorBoard(log_dir=log_dir)
```

In [41]:
```
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,1662)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
```

```
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the c
riteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_1 will not use cuDNN kernels since it doesn't meet the
criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_2 will not use cuDNN kernels since it doesn't meet the
criteria. It will use a generic GPU kernel as fallback when running on GPU.
```

In [42]:
```
model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categori
```

In [43]:
```
model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback])
```

```
Epoch 1/2000
3/3 [==============================] - 5s 306ms/step - loss: 4.2116 - categorical_ac
curacy: 0.2941
Epoch 2/2000
3/3 [==============================] - 1s 297ms/step - loss: 9.1979 - categorical_ac
curacy: 0.2824
Epoch 3/2000
3/3 [==============================] - 1s 265ms/step - loss: 16.6195 - categorical_a
ccuracy: 0.2000
Epoch 4/2000
3/3 [==============================] - 1s 252ms/step - loss: 5.8562 - categorical_ac
curacy: 0.3529
Epoch 5/2000
3/3 [==============================] - 1s 262ms/step - loss: 6.4436 - categorical_ac
curacy: 0.2471
Epoch 6/2000
3/3 [==============================] - 1s 276ms/step - loss: 8.3680 - categorical_ac
curacy: 0.2118
Epoch 7/2000
3/3 [==============================] - 1s 277ms/step - loss: 18.0270 - categorical_a
ccuracy: 0.2824
Epoch 8/2000
3/3 [==============================] - 1s 280ms/step - loss: 15.1617 - categorical_a
ccuracy: 0.2824
Epoch 9/2000
3/3 [==============================] - 1s 304ms/step - loss: 12.2953 - categorical_a
ccuracy: 0.2353
Epoch 10/2000
3/3 [==============================] - 1s 281ms/step - loss: 25.7271 - categorical_a
ccuracy: 0.2941
Epoch 11/2000
3/3 [==============================] - 1s 319ms/step - loss: 11.2907 - categorical_a
ccuracy: 0.4353
Epoch 12/2000
3/3 [==============================] - 1s 288ms/step - loss: 13.7241 - categorical_a
ccuracy: 0.4471
Epoch 13/2000
3/3 [==============================] - 1s 271ms/step - loss: 15.3459 - categorical_a
ccuracy: 0.3176
Epoch 14/2000
3/3 [==============================] - 1s 256ms/step - loss: 28.0131 - categorical_a
ccuracy: 0.3529
Epoch 15/2000
3/3 [==============================] - 1s 278ms/step - loss: 74.5101 - categorical_a
ccuracy: 0.3529
Epoch 16/2000
3/3 [==============================] - 1s 271ms/step - loss: 93.4223 - categorical_a
ccuracy: 0.3059
Epoch 17/2000
3/3 [==============================] - 1s 249ms/step - loss: 85.0892 - categorical_a
ccuracy: 0.3412
Epoch 18/2000
3/3 [==============================] - 1s 281ms/step - loss: 65.4387 - categorical_a
ccuracy: 0.3412
Epoch 19/2000
3/3 [==============================] - 1s 318ms/step - loss: 46.4009 - categorical_a
```

```
ccuracy: 0.3176
Epoch 20/2000
3/3 [==============================] - 1s 254ms/step - loss: 36.0548 - categorical_a
ccuracy: 0.3647
Epoch 21/2000
3/3 [==============================] - 1s 259ms/step - loss: 39.7644 - categorical_a
ccuracy: 0.2941
Epoch 22/2000
3/3 [==============================] - 1s 263ms/step - loss: 28.0826 - categorical_a
ccuracy: 0.3882
Epoch 23/2000
3/3 [==============================] - 1s 355ms/step - loss: 29.9280 - categorical_a
ccuracy: 0.2588
Epoch 24/2000
3/3 [==============================] - 1s 243ms/step - loss: 35.7849 - categorical_a
ccuracy: 0.3647
Epoch 25/2000
3/3 [==============================] - 1s 247ms/step - loss: 44.6540 - categorical_a
ccuracy: 0.3176
Epoch 26/2000
3/3 [==============================] - 1s 301ms/step - loss: 39.5402 - categorical_a
ccuracy: 0.3294
Epoch 27/2000
3/3 [==============================] - 1s 289ms/step - loss: 46.3419 - categorical_a
ccuracy: 0.2941
Epoch 28/2000
3/3 [==============================] - 1s 248ms/step - loss: 55.9485 - categorical_a
ccuracy: 0.2941
Epoch 29/2000
3/3 [==============================] - 1s 262ms/step - loss: 42.4990 - categorical_a
ccuracy: 0.2824
Epoch 30/2000
3/3 [==============================] - 1s 247ms/step - loss: 38.4931 - categorical_a
ccuracy: 0.3647
Epoch 31/2000
3/3 [==============================] - 1s 301ms/step - loss: 41.2612 - categorical_a
ccuracy: 0.3765
Epoch 32/2000
3/3 [==============================] - 1s 309ms/step - loss: 38.8316 - categorical_a
ccuracy: 0.3059
Epoch 33/2000
3/3 [==============================] - 1s 264ms/step - loss: 22.4168 - categorical_a
ccuracy: 0.2235
Epoch 34/2000
3/3 [==============================] - 1s 259ms/step - loss: 19.9745 - categorical_a
ccuracy: 0.3176
Epoch 35/2000
3/3 [==============================] - 1s 287ms/step - loss: 38.9197 - categorical_a
ccuracy: 0.2824
Epoch 36/2000
3/3 [==============================] - 1s 293ms/step - loss: 28.9860 - categorical_a
ccuracy: 0.4235
Epoch 37/2000
3/3 [==============================] - 1s 262ms/step - loss: 43.5768 - categorical_a
ccuracy: 0.3059
Epoch 38/2000
```

```
3/3 [==============================] - 1s 274ms/step - loss: 59.1180 - categorical_a
ccuracy: 0.3412
Epoch 39/2000
3/3 [==============================] - 1s 294ms/step - loss: 24.0920 - categorical_a
ccuracy: 0.1882
Epoch 40/2000
3/3 [==============================] - 1s 256ms/step - loss: 35.8525 - categorical_a
ccuracy: 0.2706
Epoch 41/2000
3/3 [==============================] - 1s 245ms/step - loss: 66.4587 - categorical_a
ccuracy: 0.3529
Epoch 42/2000
3/3 [==============================] - 1s 323ms/step - loss: 81.9626 - categorical_a
ccuracy: 0.1882
Epoch 43/2000
3/3 [==============================] - 1s 266ms/step - loss: 35.6864 - categorical_a
ccuracy: 0.2941
Epoch 44/2000
3/3 [==============================] - 1s 279ms/step - loss: 19.2545 - categorical_a
ccuracy: 0.4706
Epoch 45/2000
3/3 [==============================] - 1s 301ms/step - loss: 19.4613 - categorical_a
ccuracy: 0.2000
Epoch 46/2000
3/3 [==============================] - 1s 309ms/step - loss: 8.7605 - categorical_ac
curacy: 0.3765
Epoch 47/2000
3/3 [==============================] - 1s 275ms/step - loss: 4.0298 - categorical_ac
curacy: 0.5647
Epoch 48/2000
3/3 [==============================] - 1s 283ms/step - loss: 5.0213 - categorical_ac
curacy: 0.5882
Epoch 49/2000
3/3 [==============================] - 1s 267ms/step - loss: 2.6892 - categorical_ac
curacy: 0.6353
Epoch 50/2000
3/3 [==============================] - 1s 295ms/step - loss: 2.2111 - categorical_ac
curacy: 0.6471
Epoch 51/2000
3/3 [==============================] - 1s 302ms/step - loss: 4.0998 - categorical_ac
curacy: 0.5647
Epoch 52/2000
3/3 [==============================] - 1s 351ms/step - loss: 4.3993 - categorical_ac
curacy: 0.4118
Epoch 53/2000
3/3 [==============================] - 1s 279ms/step - loss: 3.8619 - categorical_ac
curacy: 0.4824
Epoch 54/2000
3/3 [==============================] - 1s 239ms/step - loss: 2.3813 - categorical_ac
curacy: 0.4706
Epoch 55/2000
3/3 [==============================] - 1s 249ms/step - loss: 2.3241 - categorical_ac
curacy: 0.4706
Epoch 56/2000
3/3 [==============================] - 1s 280ms/step - loss: 2.1600 - categorical_ac
curacy: 0.5529
```

```
Epoch 57/2000
3/3 [==============================] - 1s 247ms/step - loss: 3.1088 - categorical_ac
curacy: 0.4824
Epoch 58/2000
3/3 [==============================] - 1s 262ms/step - loss: 1.5071 - categorical_ac
curacy: 0.5882
Epoch 59/2000
3/3 [==============================] - 1s 249ms/step - loss: 1.5778 - categorical_ac
curacy: 0.5765
Epoch 60/2000
3/3 [==============================] - 1s 249ms/step - loss: 1.3208 - categorical_ac
curacy: 0.6235
Epoch 61/2000
3/3 [==============================] - 1s 262ms/step - loss: 1.2427 - categorical_ac
curacy: 0.6471
Epoch 62/2000
3/3 [==============================] - 1s 248ms/step - loss: 1.3204 - categorical_ac
curacy: 0.6471
Epoch 63/2000
3/3 [==============================] - 1s 259ms/step - loss: 1.4067 - categorical_ac
curacy: 0.6235
Epoch 64/2000
3/3 [==============================] - 1s 314ms/step - loss: 1.4068 - categorical_ac
curacy: 0.6235
Epoch 65/2000
3/3 [==============================] - 1s 244ms/step - loss: 1.9098 - categorical_ac
curacy: 0.6588
Epoch 66/2000
3/3 [==============================] - 1s 267ms/step - loss: 1.9888 - categorical_ac
curacy: 0.6353
Epoch 67/2000
3/3 [==============================] - 1s 286ms/step - loss: 2.1508 - categorical_ac
curacy: 0.6471
Epoch 68/2000
3/3 [==============================] - 1s 245ms/step - loss: 1.3949 - categorical_ac
curacy: 0.6588
Epoch 69/2000
3/3 [==============================] - 1s 282ms/step - loss: 1.3748 - categorical_ac
curacy: 0.6824
Epoch 70/2000
3/3 [==============================] - 1s 319ms/step - loss: 1.0419 - categorical_ac
curacy: 0.6706
Epoch 71/2000
3/3 [==============================] - 1s 289ms/step - loss: 1.1619 - categorical_ac
curacy: 0.7059
Epoch 72/2000
3/3 [==============================] - 1s 253ms/step - loss: 0.9958 - categorical_ac
curacy: 0.6706
Epoch 73/2000
3/3 [==============================] - 1s 259ms/step - loss: 1.4235 - categorical_ac
curacy: 0.6471
Epoch 74/2000
3/3 [==============================] - 1s 250ms/step - loss: 1.1423 - categorical_ac
curacy: 0.6588
Epoch 75/2000
3/3 [==============================] - 1s 246ms/step - loss: 1.8165 - categorical_ac
```

```
curacy: 0.6471
Epoch 76/2000
3/3 [==============================] - 1s 277ms/step - loss: 2.9078 - categorical_ac
curacy: 0.6588
Epoch 77/2000
3/3 [==============================] - 1s 290ms/step - loss: 1.7519 - categorical_ac
curacy: 0.7059
Epoch 78/2000
3/3 [==============================] - 1s 252ms/step - loss: 1.3561 - categorical_ac
curacy: 0.6471
Epoch 79/2000
3/3 [==============================] - 1s 270ms/step - loss: 1.6979 - categorical_ac
curacy: 0.6118
Epoch 80/2000
3/3 [==============================] - 1s 279ms/step - loss: 0.9905 - categorical_ac
curacy: 0.6471
Epoch 81/2000
3/3 [==============================] - 1s 248ms/step - loss: 1.1560 - categorical_ac
curacy: 0.6706
Epoch 82/2000
3/3 [==============================] - 1s 251ms/step - loss: 1.1212 - categorical_ac
curacy: 0.6706
Epoch 83/2000
3/3 [==============================] - 1s 283ms/step - loss: 0.7902 - categorical_ac
curacy: 0.7176
Epoch 84/2000
3/3 [==============================] - 1s 240ms/step - loss: 0.9226 - categorical_ac
curacy: 0.7059
Epoch 85/2000
3/3 [==============================] - 1s 256ms/step - loss: 0.9543 - categorical_ac
curacy: 0.6824
Epoch 86/2000
3/3 [==============================] - 1s 286ms/step - loss: 1.7899 - categorical_ac
curacy: 0.6941
Epoch 87/2000
3/3 [==============================] - 1s 309ms/step - loss: 1.5108 - categorical_ac
curacy: 0.6471
Epoch 88/2000
3/3 [==============================] - 1s 291ms/step - loss: 1.4358 - categorical_ac
curacy: 0.6353
Epoch 89/2000
3/3 [==============================] - 1s 334ms/step - loss: 1.0738 - categorical_ac
curacy: 0.6588
Epoch 90/2000
3/3 [==============================] - 1s 334ms/step - loss: 0.9967 - categorical_ac
curacy: 0.7294
Epoch 91/2000
3/3 [==============================] - 1s 332ms/step - loss: 0.8015 - categorical_ac
curacy: 0.7294
Epoch 92/2000
3/3 [==============================] - 1s 316ms/step - loss: 0.8170 - categorical_ac
curacy: 0.7294
Epoch 93/2000
3/3 [==============================] - 1s 295ms/step - loss: 0.7920 - categorical_ac
curacy: 0.7529
Epoch 94/2000
```

```
3/3 [==============================] - 1s 251ms/step - loss: 1.2964 - categorical_ac
curacy: 0.6471
Epoch 95/2000
3/3 [==============================] - 1s 317ms/step - loss: 2.4268 - categorical_ac
curacy: 0.6588
Epoch 96/2000
3/3 [==============================] - 1s 252ms/step - loss: 2.2036 - categorical_ac
curacy: 0.6706
Epoch 97/2000
3/3 [==============================] - 1s 255ms/step - loss: 1.4681 - categorical_ac
curacy: 0.6824
Epoch 98/2000
3/3 [==============================] - 1s 348ms/step - loss: 1.3076 - categorical_ac
curacy: 0.7529
Epoch 99/2000
3/3 [==============================] - 1s 271ms/step - loss: 2.5215 - categorical_ac
curacy: 0.6235
Epoch 100/2000
3/3 [==============================] - 1s 267ms/step - loss: 1.7688 - categorical_ac
curacy: 0.6000
Epoch 101/2000
3/3 [==============================] - 1s 335ms/step - loss: 0.9716 - categorical_ac
curacy: 0.6000
Epoch 102/2000
3/3 [==============================] - 1s 295ms/step - loss: 1.5549 - categorical_ac
curacy: 0.6000
Epoch 103/2000
3/3 [==============================] - 1s 246ms/step - loss: 1.1546 - categorical_ac
curacy: 0.6235
Epoch 104/2000
3/3 [==============================] - 1s 247ms/step - loss: 1.1501 - categorical_ac
curacy: 0.7059
Epoch 105/2000
3/3 [==============================] - 1s 317ms/step - loss: 1.3162 - categorical_ac
curacy: 0.7059
Epoch 106/2000
3/3 [==============================] - 1s 303ms/step - loss: 1.5962 - categorical_ac
curacy: 0.6235
Epoch 107/2000
3/3 [==============================] - 1s 245ms/step - loss: 1.1337 - categorical_ac
curacy: 0.6471
Epoch 108/2000
3/3 [==============================] - 1s 279ms/step - loss: 1.0024 - categorical_ac
curacy: 0.7059
Epoch 109/2000
3/3 [==============================] - 1s 337ms/step - loss: 0.8914 - categorical_ac
curacy: 0.5882
Epoch 110/2000
3/3 [==============================] - 1s 300ms/step - loss: 0.7824 - categorical_ac
curacy: 0.7294
Epoch 111/2000
3/3 [==============================] - 1s 300ms/step - loss: 0.8106 - categorical_ac
curacy: 0.7529
Epoch 112/2000
3/3 [==============================] - 1s 288ms/step - loss: 0.7524 - categorical_ac
curacy: 0.6824
```

```
Epoch 113/2000
3/3 [==============================] - 1s 266ms/step - loss: 0.8532 - categorical_ac
curacy: 0.7059
Epoch 114/2000
3/3 [==============================] - 1s 292ms/step - loss: 0.6567 - categorical_ac
curacy: 0.7059
Epoch 115/2000
3/3 [==============================] - 1s 313ms/step - loss: 0.8151 - categorical_ac
curacy: 0.7176
Epoch 116/2000
3/3 [==============================] - 1s 331ms/step - loss: 1.2592 - categorical_ac
curacy: 0.6235
Epoch 117/2000
3/3 [==============================] - 1s 265ms/step - loss: 0.5053 - categorical_ac
curacy: 0.7765
Epoch 118/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.7285 - categorical_ac
curacy: 0.7294
Epoch 119/2000
3/3 [==============================] - 1s 297ms/step - loss: 0.5644 - categorical_ac
curacy: 0.7647
Epoch 120/2000
3/3 [==============================] - 1s 317ms/step - loss: 0.6121 - categorical_ac
curacy: 0.7412
Epoch 121/2000
3/3 [==============================] - 1s 241ms/step - loss: 0.6667 - categorical_ac
curacy: 0.6941
Epoch 122/2000
3/3 [==============================] - 1s 256ms/step - loss: 0.6365 - categorical_ac
curacy: 0.7529
Epoch 123/2000
3/3 [==============================] - 1s 346ms/step - loss: 0.5573 - categorical_ac
curacy: 0.6941
Epoch 124/2000
3/3 [==============================] - 1s 246ms/step - loss: 1.0077 - categorical_ac
curacy: 0.7059
Epoch 125/2000
3/3 [==============================] - 1s 243ms/step - loss: 1.2524 - categorical_ac
curacy: 0.6824
Epoch 126/2000
3/3 [==============================] - 1s 273ms/step - loss: 1.0672 - categorical_ac
curacy: 0.6941
Epoch 127/2000
3/3 [==============================] - 1s 294ms/step - loss: 1.0558 - categorical_ac
curacy: 0.6941
Epoch 128/2000
3/3 [==============================] - 1s 336ms/step - loss: 0.4332 - categorical_ac
curacy: 0.8353
Epoch 129/2000
3/3 [==============================] - 1s 263ms/step - loss: 0.3791 - categorical_ac
curacy: 0.8471
Epoch 130/2000
3/3 [==============================] - 1s 287ms/step - loss: 0.4075 - categorical_ac
curacy: 0.8588
Epoch 131/2000
3/3 [==============================] - 1s 246ms/step - loss: 0.4094 - categorical_ac
```

```
curacy: 0.8353
Epoch 132/2000
3/3 [==============================] - 1s 247ms/step - loss: 0.4854 - categorical_ac
curacy: 0.8353
Epoch 133/2000
3/3 [==============================] - 1s 271ms/step - loss: 0.4537 - categorical_ac
curacy: 0.8235
Epoch 134/2000
3/3 [==============================] - 1s 290ms/step - loss: 0.3361 - categorical_ac
curacy: 0.8824
Epoch 135/2000
3/3 [==============================] - 1s 295ms/step - loss: 0.3512 - categorical_ac
curacy: 0.8941
Epoch 136/2000
3/3 [==============================] - 1s 291ms/step - loss: 0.4028 - categorical_ac
curacy: 0.8235
Epoch 137/2000
3/3 [==============================] - 1s 281ms/step - loss: 0.4315 - categorical_ac
curacy: 0.8118
Epoch 138/2000
3/3 [==============================] - 1s 244ms/step - loss: 0.6040 - categorical_ac
curacy: 0.7412
Epoch 139/2000
3/3 [==============================] - 1s 270ms/step - loss: 0.5400 - categorical_ac
curacy: 0.7765
Epoch 140/2000
3/3 [==============================] - 1s 248ms/step - loss: 0.7295 - categorical_ac
curacy: 0.7412
Epoch 141/2000
3/3 [==============================] - 1s 283ms/step - loss: 0.7063 - categorical_ac
curacy: 0.7412
Epoch 142/2000
3/3 [==============================] - 1s 301ms/step - loss: 0.7234 - categorical_ac
curacy: 0.6588
Epoch 143/2000
3/3 [==============================] - 1s 260ms/step - loss: 0.5099 - categorical_ac
curacy: 0.7647
Epoch 144/2000
3/3 [==============================] - 1s 248ms/step - loss: 0.5270 - categorical_ac
curacy: 0.8235
Epoch 145/2000
3/3 [==============================] - 1s 263ms/step - loss: 0.3454 - categorical_ac
curacy: 0.8588
Epoch 146/2000
3/3 [==============================] - 1s 281ms/step - loss: 0.3431 - categorical_ac
curacy: 0.8706
Epoch 147/2000
3/3 [==============================] - 1s 277ms/step - loss: 0.3842 - categorical_ac
curacy: 0.8471
Epoch 148/2000
3/3 [==============================] - 1s 270ms/step - loss: 0.3122 - categorical_ac
curacy: 0.8588
Epoch 149/2000
3/3 [==============================] - 1s 260ms/step - loss: 0.3217 - categorical_ac
curacy: 0.8588
Epoch 150/2000
```

```
3/3 [==============================] - 1s 256ms/step - loss: 0.2977 - categorical_ac
curacy: 0.8941
Epoch 151/2000
3/3 [==============================] - 1s 273ms/step - loss: 0.3083 - categorical_ac
curacy: 0.8824
Epoch 152/2000
3/3 [==============================] - 1s 269ms/step - loss: 0.2889 - categorical_ac
curacy: 0.8706
Epoch 153/2000
3/3 [==============================] - 1s 244ms/step - loss: 0.2936 - categorical_ac
curacy: 0.8706
Epoch 154/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.3391 - categorical_ac
curacy: 0.8235
Epoch 155/2000
3/3 [==============================] - 1s 264ms/step - loss: 0.3604 - categorical_ac
curacy: 0.8353
Epoch 156/2000
3/3 [==============================] - 1s 245ms/step - loss: 0.4375 - categorical_ac
curacy: 0.7765
Epoch 157/2000
3/3 [==============================] - 1s 309ms/step - loss: 0.3498 - categorical_ac
curacy: 0.8235
Epoch 158/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.3296 - categorical_ac
curacy: 0.8824
Epoch 159/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.2905 - categorical_ac
curacy: 0.8588
Epoch 160/2000
3/3 [==============================] - 1s 283ms/step - loss: 0.2633 - categorical_ac
curacy: 0.8941
Epoch 161/2000
3/3 [==============================] - 1s 284ms/step - loss: 0.2854 - categorical_ac
curacy: 0.8588
Epoch 162/2000
3/3 [==============================] - 1s 265ms/step - loss: 0.3888 - categorical_ac
curacy: 0.8706
Epoch 163/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.3194 - categorical_ac
curacy: 0.8118
Epoch 164/2000
3/3 [==============================] - 1s 323ms/step - loss: 0.5341 - categorical_ac
curacy: 0.7412
Epoch 165/2000
3/3 [==============================] - 1s 376ms/step - loss: 0.5773 - categorical_ac
curacy: 0.7412
Epoch 166/2000
3/3 [==============================] - 1s 317ms/step - loss: 0.6204 - categorical_ac
curacy: 0.7412
Epoch 167/2000
3/3 [==============================] - 1s 303ms/step - loss: 0.5218 - categorical_ac
curacy: 0.7765
Epoch 168/2000
3/3 [==============================] - 1s 312ms/step - loss: 0.5332 - categorical_ac
curacy: 0.7765
```

```
Epoch 169/2000
3/3 [==============================] - 1s 267ms/step - loss: 0.5250 - categorical_ac
curacy: 0.7529
Epoch 170/2000
3/3 [==============================] - 1s 273ms/step - loss: 0.7499 - categorical_ac
curacy: 0.6941
Epoch 171/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.6777 - categorical_ac
curacy: 0.7412
Epoch 172/2000
3/3 [==============================] - 1s 303ms/step - loss: 1.0698 - categorical_ac
curacy: 0.7176
Epoch 173/2000
3/3 [==============================] - 1s 256ms/step - loss: 1.4508 - categorical_ac
curacy: 0.6706
Epoch 174/2000
3/3 [==============================] - 1s 244ms/step - loss: 0.8741 - categorical_ac
curacy: 0.7294
Epoch 175/2000
3/3 [==============================] - 1s 324ms/step - loss: 0.5082 - categorical_ac
curacy: 0.7765
Epoch 176/2000
3/3 [==============================] - 1s 259ms/step - loss: 0.4151 - categorical_ac
curacy: 0.8235
Epoch 177/2000
3/3 [==============================] - 1s 269ms/step - loss: 0.4184 - categorical_ac
curacy: 0.7882
Epoch 178/2000
3/3 [==============================] - 1s 258ms/step - loss: 0.3840 - categorical_ac
curacy: 0.8235
Epoch 179/2000
3/3 [==============================] - 1s 271ms/step - loss: 0.3581 - categorical_ac
curacy: 0.8588
Epoch 180/2000
3/3 [==============================] - 1s 243ms/step - loss: 0.4786 - categorical_ac
curacy: 0.8000
Epoch 181/2000
3/3 [==============================] - 1s 248ms/step - loss: 0.3980 - categorical_ac
curacy: 0.8000
Epoch 182/2000
3/3 [==============================] - 1s 294ms/step - loss: 0.3953 - categorical_ac
curacy: 0.8000
Epoch 183/2000
3/3 [==============================] - 1s 243ms/step - loss: 0.5958 - categorical_ac
curacy: 0.6941
Epoch 184/2000
3/3 [==============================] - 1s 295ms/step - loss: 0.3896 - categorical_ac
curacy: 0.7647
Epoch 185/2000
3/3 [==============================] - 1s 392ms/step - loss: 0.4232 - categorical_ac
curacy: 0.8235
Epoch 186/2000
3/3 [==============================] - 1s 277ms/step - loss: 0.4076 - categorical_ac
curacy: 0.8000
Epoch 187/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.3388 - categorical_ac
```

```
curacy: 0.8118
Epoch 188/2000
3/3 [==============================] - 1s 263ms/step - loss: 0.3282 - categorical_ac
curacy: 0.8706
Epoch 189/2000
3/3 [==============================] - 1s 269ms/step - loss: 0.3233 - categorical_ac
curacy: 0.8235
Epoch 190/2000
3/3 [==============================] - 1s 290ms/step - loss: 0.3358 - categorical_ac
curacy: 0.8471
Epoch 191/2000
3/3 [==============================] - 1s 284ms/step - loss: 0.3074 - categorical_ac
curacy: 0.8706
Epoch 192/2000
3/3 [==============================] - 1s 269ms/step - loss: 0.3084 - categorical_ac
curacy: 0.8353
Epoch 193/2000
3/3 [==============================] - 1s 267ms/step - loss: 0.6150 - categorical_ac
curacy: 0.6941
Epoch 194/2000
3/3 [==============================] - 1s 280ms/step - loss: 0.3670 - categorical_ac
curacy: 0.8118
Epoch 195/2000
3/3 [==============================] - 1s 324ms/step - loss: 0.4525 - categorical_ac
curacy: 0.8235
Epoch 196/2000
3/3 [==============================] - 1s 281ms/step - loss: 0.4013 - categorical_ac
curacy: 0.8235
Epoch 197/2000
3/3 [==============================] - 1s 264ms/step - loss: 0.3984 - categorical_ac
curacy: 0.7882
Epoch 198/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.2719 - categorical_ac
curacy: 0.8706
Epoch 199/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.3798 - categorical_ac
curacy: 0.8000
Epoch 200/2000
3/3 [==============================] - 1s 313ms/step - loss: 0.3218 - categorical_ac
curacy: 0.8824
Epoch 201/2000
3/3 [==============================] - 1s 352ms/step - loss: 0.3629 - categorical_ac
curacy: 0.8000
Epoch 202/2000
3/3 [==============================] - 1s 270ms/step - loss: 0.3889 - categorical_ac
curacy: 0.8235
Epoch 203/2000
3/3 [==============================] - 1s 312ms/step - loss: 0.8279 - categorical_ac
curacy: 0.6824
Epoch 204/2000
3/3 [==============================] - 1s 334ms/step - loss: 0.7156 - categorical_ac
curacy: 0.7059
Epoch 205/2000
3/3 [==============================] - 1s 298ms/step - loss: 0.3270 - categorical_ac
curacy: 0.8353
Epoch 206/2000
```

```
3/3 [==============================] - 1s 305ms/step - loss: 0.4112 - categorical_ac
curacy: 0.7882
Epoch 207/2000
3/3 [==============================] - 1s 287ms/step - loss: 0.5313 - categorical_ac
curacy: 0.7176
Epoch 208/2000
3/3 [==============================] - 1s 288ms/step - loss: 0.9790 - categorical_ac
curacy: 0.7176
Epoch 209/2000
3/3 [==============================] - 1s 265ms/step - loss: 0.4739 - categorical_ac
curacy: 0.8000
Epoch 210/2000
3/3 [==============================] - 1s 262ms/step - loss: 0.3640 - categorical_ac
curacy: 0.8353
Epoch 211/2000
3/3 [==============================] - 1s 271ms/step - loss: 0.4012 - categorical_ac
curacy: 0.7882
Epoch 212/2000
3/3 [==============================] - 1s 266ms/step - loss: 0.3455 - categorical_ac
curacy: 0.8588
Epoch 213/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.3345 - categorical_ac
curacy: 0.8235
Epoch 214/2000
3/3 [==============================] - 1s 279ms/step - loss: 0.3495 - categorical_ac
curacy: 0.8353
Epoch 215/2000
3/3 [==============================] - 1s 253ms/step - loss: 0.4073 - categorical_ac
curacy: 0.8118
Epoch 216/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.7585 - categorical_ac
curacy: 0.7529
Epoch 217/2000
3/3 [==============================] - 1s 270ms/step - loss: 1.0403 - categorical_ac
curacy: 0.6941
Epoch 218/2000
3/3 [==============================] - 1s 275ms/step - loss: 0.9646 - categorical_ac
curacy: 0.6706
Epoch 219/2000
3/3 [==============================] - 1s 256ms/step - loss: 0.3220 - categorical_ac
curacy: 0.9176
Epoch 220/2000
3/3 [==============================] - 1s 262ms/step - loss: 0.6064 - categorical_ac
curacy: 0.7882
Epoch 221/2000
3/3 [==============================] - 1s 294ms/step - loss: 0.3481 - categorical_ac
curacy: 0.8588
Epoch 222/2000
3/3 [==============================] - 1s 281ms/step - loss: 0.4856 - categorical_ac
curacy: 0.8118
Epoch 223/2000
3/3 [==============================] - 1s 274ms/step - loss: 0.4371 - categorical_ac
curacy: 0.8353
Epoch 224/2000
3/3 [==============================] - 1s 283ms/step - loss: 0.4842 - categorical_ac
curacy: 0.8588
```

```
Epoch 225/2000
3/3 [==============================] - 1s 248ms/step - loss: 0.6178 - categorical_ac
curacy: 0.7647
Epoch 226/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.6138 - categorical_ac
curacy: 0.7647
Epoch 227/2000
3/3 [==============================] - 1s 291ms/step - loss: 0.7997 - categorical_ac
curacy: 0.7529
Epoch 228/2000
3/3 [==============================] - 1s 247ms/step - loss: 0.5936 - categorical_ac
curacy: 0.7765
Epoch 229/2000
3/3 [==============================] - 1s 255ms/step - loss: 0.5765 - categorical_ac
curacy: 0.8235
Epoch 230/2000
3/3 [==============================] - 1s 323ms/step - loss: 0.7067 - categorical_ac
curacy: 0.7529
Epoch 231/2000
3/3 [==============================] - 1s 293ms/step - loss: 0.7527 - categorical_ac
curacy: 0.7529
Epoch 232/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.6108 - categorical_ac
curacy: 0.8118
Epoch 233/2000
3/3 [==============================] - 1s 255ms/step - loss: 0.4577 - categorical_ac
curacy: 0.8471
Epoch 234/2000
3/3 [==============================] - 1s 283ms/step - loss: 0.3028 - categorical_ac
curacy: 0.8706
Epoch 235/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.3184 - categorical_ac
curacy: 0.8706
Epoch 236/2000
3/3 [==============================] - 1s 338ms/step - loss: 0.3496 - categorical_ac
curacy: 0.8353
Epoch 237/2000
3/3 [==============================] - 1s 351ms/step - loss: 0.5598 - categorical_ac
curacy: 0.8235
Epoch 238/2000
3/3 [==============================] - 1s 264ms/step - loss: 0.4843 - categorical_ac
curacy: 0.8235
Epoch 239/2000
3/3 [==============================] - 1s 276ms/step - loss: 0.5535 - categorical_ac
curacy: 0.7882
Epoch 240/2000
3/3 [==============================] - 1s 336ms/step - loss: 0.7112 - categorical_ac
curacy: 0.7647
Epoch 241/2000
3/3 [==============================] - 1s 357ms/step - loss: 0.3342 - categorical_ac
curacy: 0.8706
Epoch 242/2000
3/3 [==============================] - 1s 341ms/step - loss: 0.3018 - categorical_ac
curacy: 0.8706
Epoch 243/2000
3/3 [==============================] - 1s 307ms/step - loss: 0.3400 - categorical_ac
```

```
curacy: 0.8353
Epoch 244/2000
3/3 [==============================] - 1s 279ms/step - loss: 0.3655 - categorical_ac
curacy: 0.8471
Epoch 245/2000
3/3 [==============================] - 1s 311ms/step - loss: 0.4638 - categorical_ac
curacy: 0.8353
Epoch 246/2000
3/3 [==============================] - 1s 270ms/step - loss: 0.3843 - categorical_ac
curacy: 0.8706
Epoch 247/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.2975 - categorical_ac
curacy: 0.8588
Epoch 248/2000
3/3 [==============================] - 1s 258ms/step - loss: 0.5884 - categorical_ac
curacy: 0.7882
Epoch 249/2000
3/3 [==============================] - 1s 257ms/step - loss: 0.3943 - categorical_ac
curacy: 0.8353
Epoch 250/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.3574 - categorical_ac
curacy: 0.8353
Epoch 251/2000
3/3 [==============================] - 1s 339ms/step - loss: 0.3494 - categorical_ac
curacy: 0.8706
Epoch 252/2000
3/3 [==============================] - 1s 316ms/step - loss: 0.2917 - categorical_ac
curacy: 0.8824
Epoch 253/2000
3/3 [==============================] - 1s 255ms/step - loss: 0.3499 - categorical_ac
curacy: 0.8235
Epoch 254/2000
3/3 [==============================] - 1s 281ms/step - loss: 0.5496 - categorical_ac
curacy: 0.7882
Epoch 255/2000
3/3 [==============================] - 1s 327ms/step - loss: 0.3657 - categorical_ac
curacy: 0.8706
Epoch 256/2000
3/3 [==============================] - 1s 321ms/step - loss: 0.3102 - categorical_ac
curacy: 0.8471
Epoch 257/2000
3/3 [==============================] - 1s 298ms/step - loss: 0.3413 - categorical_ac
curacy: 0.8588
Epoch 258/2000
3/3 [==============================] - 1s 250ms/step - loss: 0.2587 - categorical_ac
curacy: 0.8941
Epoch 259/2000
3/3 [==============================] - 1s 313ms/step - loss: 0.3392 - categorical_ac
curacy: 0.8588
Epoch 260/2000
3/3 [==============================] - 1s 277ms/step - loss: 0.3427 - categorical_ac
curacy: 0.8471
Epoch 261/2000
3/3 [==============================] - 1s 270ms/step - loss: 0.3734 - categorical_ac
curacy: 0.8471
Epoch 262/2000
```

```
3/3 [==============================] - 1s 295ms/step - loss: 0.2011 - categorical_ac
curacy: 0.8941
Epoch 263/2000
3/3 [==============================] - 1s 258ms/step - loss: 0.2559 - categorical_ac
curacy: 0.8588
Epoch 264/2000
3/3 [==============================] - 1s 241ms/step - loss: 0.2452 - categorical_ac
curacy: 0.9176
Epoch 265/2000
3/3 [==============================] - 1s 264ms/step - loss: 0.2228 - categorical_ac
curacy: 0.8824
Epoch 266/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.1754 - categorical_ac
curacy: 0.9176
Epoch 267/2000
3/3 [==============================] - 1s 285ms/step - loss: 0.2965 - categorical_ac
curacy: 0.8706
Epoch 268/2000
3/3 [==============================] - 1s 291ms/step - loss: 0.3849 - categorical_ac
curacy: 0.8118
Epoch 269/2000
3/3 [==============================] - 1s 271ms/step - loss: 0.3860 - categorical_ac
curacy: 0.8000
Epoch 270/2000
3/3 [==============================] - 1s 295ms/step - loss: 0.3906 - categorical_ac
curacy: 0.8353
Epoch 271/2000
3/3 [==============================] - 1s 333ms/step - loss: 0.4213 - categorical_ac
curacy: 0.8353
Epoch 272/2000
3/3 [==============================] - 1s 321ms/step - loss: 0.6585 - categorical_ac
curacy: 0.7765
Epoch 273/2000
3/3 [==============================] - 1s 264ms/step - loss: 0.9811 - categorical_ac
curacy: 0.7294
Epoch 274/2000
3/3 [==============================] - 1s 252ms/step - loss: 0.8109 - categorical_ac
curacy: 0.7529
Epoch 275/2000
3/3 [==============================] - 1s 302ms/step - loss: 0.4514 - categorical_ac
curacy: 0.8235
Epoch 276/2000
3/3 [==============================] - 1s 289ms/step - loss: 0.5387 - categorical_ac
curacy: 0.7765
Epoch 277/2000
3/3 [==============================] - 1s 330ms/step - loss: 0.5455 - categorical_ac
curacy: 0.7882
Epoch 278/2000
3/3 [==============================] - 1s 298ms/step - loss: 0.6582 - categorical_ac
curacy: 0.7647
Epoch 279/2000
3/3 [==============================] - 1s 340ms/step - loss: 0.2823 - categorical_ac
curacy: 0.8588
Epoch 280/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.3661 - categorical_ac
curacy: 0.8235
```

```
Epoch 281/2000
3/3 [==============================] - 1s 292ms/step - loss: 0.3576 - categorical_ac
curacy: 0.8588
Epoch 282/2000
3/3 [==============================] - 1s 364ms/step - loss: 0.3609 - categorical_ac
curacy: 0.8353
Epoch 283/2000
3/3 [==============================] - 1s 295ms/step - loss: 0.6411 - categorical_ac
curacy: 0.7647
Epoch 284/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.3164 - categorical_ac
curacy: 0.8471
Epoch 285/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.4269 - categorical_ac
curacy: 0.8353
Epoch 286/2000
3/3 [==============================] - 1s 313ms/step - loss: 0.5717 - categorical_ac
curacy: 0.7647
Epoch 287/2000
3/3 [==============================] - 1s 260ms/step - loss: 0.8156 - categorical_ac
curacy: 0.7765
Epoch 288/2000
3/3 [==============================] - 1s 290ms/step - loss: 0.8419 - categorical_ac
curacy: 0.7294
Epoch 289/2000
3/3 [==============================] - 1s 336ms/step - loss: 0.7549 - categorical_ac
curacy: 0.7529
Epoch 290/2000
3/3 [==============================] - 1s 257ms/step - loss: 0.7404 - categorical_ac
curacy: 0.7647
Epoch 291/2000
3/3 [==============================] - 1s 253ms/step - loss: 0.7141 - categorical_ac
curacy: 0.8118
Epoch 292/2000
3/3 [==============================] - 1s 280ms/step - loss: 0.6033 - categorical_ac
curacy: 0.7647
Epoch 293/2000
3/3 [==============================] - 1s 271ms/step - loss: 0.6839 - categorical_ac
curacy: 0.8118
Epoch 294/2000
3/3 [==============================] - 1s 308ms/step - loss: 0.9578 - categorical_ac
curacy: 0.7412
Epoch 295/2000
3/3 [==============================] - 1s 303ms/step - loss: 0.9756 - categorical_ac
curacy: 0.7294
Epoch 296/2000
3/3 [==============================] - 1s 301ms/step - loss: 1.0519 - categorical_ac
curacy: 0.7765
Epoch 297/2000
3/3 [==============================] - 1s 301ms/step - loss: 1.0288 - categorical_ac
curacy: 0.7529
Epoch 298/2000
3/3 [==============================] - 1s 293ms/step - loss: 1.3371 - categorical_ac
curacy: 0.6824
Epoch 299/2000
3/3 [==============================] - 1s 262ms/step - loss: 0.7660 - categorical_ac
```

```
curacy: 0.7765
Epoch 300/2000
3/3 [==============================] - 1s 293ms/step - loss: 0.6826 - categorical_ac
curacy: 0.7882
Epoch 301/2000
3/3 [==============================] - 1s 340ms/step - loss: 0.7796 - categorical_ac
curacy: 0.7412
Epoch 302/2000
3/3 [==============================] - 1s 318ms/step - loss: 1.2648 - categorical_ac
curacy: 0.7294
Epoch 303/2000
3/3 [==============================] - 1s 332ms/step - loss: 0.6494 - categorical_ac
curacy: 0.7765
Epoch 304/2000
3/3 [==============================] - 1s 321ms/step - loss: 1.1562 - categorical_ac
curacy: 0.7059
Epoch 305/2000
3/3 [==============================] - 1s 292ms/step - loss: 1.8090 - categorical_ac
curacy: 0.6471
Epoch 306/2000
3/3 [==============================] - 1s 260ms/step - loss: 1.1911 - categorical_ac
curacy: 0.6824
Epoch 307/2000
3/3 [==============================] - 1s 250ms/step - loss: 1.0877 - categorical_ac
curacy: 0.7059
Epoch 308/2000
3/3 [==============================] - 1s 262ms/step - loss: 2.4544 - categorical_ac
curacy: 0.6588
Epoch 309/2000
3/3 [==============================] - 1s 351ms/step - loss: 1.5035 - categorical_ac
curacy: 0.6824
Epoch 310/2000
3/3 [==============================] - 1s 270ms/step - loss: 1.2980 - categorical_ac
curacy: 0.6824
Epoch 311/2000
3/3 [==============================] - 1s 277ms/step - loss: 0.9902 - categorical_ac
curacy: 0.6824
Epoch 312/2000
3/3 [==============================] - 1s 295ms/step - loss: 0.4431 - categorical_ac
curacy: 0.7882
Epoch 313/2000
3/3 [==============================] - 1s 282ms/step - loss: 0.5215 - categorical_ac
curacy: 0.8235
Epoch 314/2000
3/3 [==============================] - 1s 279ms/step - loss: 0.7604 - categorical_ac
curacy: 0.7765
Epoch 315/2000
3/3 [==============================] - 1s 338ms/step - loss: 1.0651 - categorical_ac
curacy: 0.7529
Epoch 316/2000
3/3 [==============================] - 1s 278ms/step - loss: 0.6945 - categorical_ac
curacy: 0.7059
Epoch 317/2000
3/3 [==============================] - 1s 298ms/step - loss: 1.3412 - categorical_ac
curacy: 0.7059
Epoch 318/2000
```

```
3/3 [==============================] - 1s 292ms/step - loss: 0.4664 - categorical_ac
curacy: 0.8000
Epoch 319/2000
3/3 [==============================] - 1s 307ms/step - loss: 0.8041 - categorical_ac
curacy: 0.7412
Epoch 320/2000
3/3 [==============================] - 1s 290ms/step - loss: 0.5504 - categorical_ac
curacy: 0.8118
Epoch 321/2000
3/3 [==============================] - 1s 287ms/step - loss: 1.0148 - categorical_ac
curacy: 0.7647
Epoch 322/2000
3/3 [==============================] - 1s 330ms/step - loss: 1.7817 - categorical_ac
curacy: 0.6235
Epoch 323/2000
3/3 [==============================] - 1s 250ms/step - loss: 1.2570 - categorical_ac
curacy: 0.7176
Epoch 324/2000
3/3 [==============================] - 1s 259ms/step - loss: 0.6285 - categorical_ac
curacy: 0.7529
Epoch 325/2000
3/3 [==============================] - 1s 346ms/step - loss: 1.1794 - categorical_ac
curacy: 0.6706
Epoch 326/2000
3/3 [==============================] - 1s 255ms/step - loss: 1.1590 - categorical_ac
curacy: 0.6471
Epoch 327/2000
3/3 [==============================] - 1s 250ms/step - loss: 1.2647 - categorical_ac
curacy: 0.6706
Epoch 328/2000
3/3 [==============================] - 1s 263ms/step - loss: 1.3814 - categorical_ac
curacy: 0.6353
Epoch 329/2000
3/3 [==============================] - 1s 359ms/step - loss: 1.0930 - categorical_ac
curacy: 0.6941
Epoch 330/2000
3/3 [==============================] - 1s 248ms/step - loss: 0.9085 - categorical_ac
curacy: 0.6353
Epoch 331/2000
3/3 [==============================] - 1s 250ms/step - loss: 1.0045 - categorical_ac
curacy: 0.7059
Epoch 332/2000
3/3 [==============================] - 1s 366ms/step - loss: 1.1278 - categorical_ac
curacy: 0.6118
Epoch 333/2000
3/3 [==============================] - 1s 356ms/step - loss: 0.7687 - categorical_ac
curacy: 0.7412
Epoch 334/2000
3/3 [==============================] - 1s 283ms/step - loss: 1.1742 - categorical_ac
curacy: 0.7059
Epoch 335/2000
3/3 [==============================] - 1s 276ms/step - loss: 1.1076 - categorical_ac
curacy: 0.7412
Epoch 336/2000
3/3 [==============================] - 1s 276ms/step - loss: 1.5232 - categorical_ac
curacy: 0.6118
```

```
Epoch 337/2000
3/3 [==============================] - 1s 286ms/step - loss: 0.9759 - categorical_ac
curacy: 0.6824
Epoch 338/2000
3/3 [==============================] - 1s 250ms/step - loss: 1.3659 - categorical_ac
curacy: 0.6118
Epoch 339/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.7657 - categorical_ac
curacy: 0.7176
Epoch 340/2000
3/3 [==============================] - 1s 344ms/step - loss: 0.8140 - categorical_ac
curacy: 0.7529
Epoch 341/2000
3/3 [==============================] - 1s 330ms/step - loss: 0.9073 - categorical_ac
curacy: 0.6588
Epoch 342/2000
3/3 [==============================] - 1s 268ms/step - loss: 1.3363 - categorical_ac
curacy: 0.6824
Epoch 343/2000
3/3 [==============================] - 1s 348ms/step - loss: 1.2815 - categorical_ac
curacy: 0.6000
Epoch 344/2000
3/3 [==============================] - 1s 340ms/step - loss: 1.1833 - categorical_ac
curacy: 0.6706
Epoch 345/2000
3/3 [==============================] - 1s 288ms/step - loss: 1.0706 - categorical_ac
curacy: 0.6706
Epoch 346/2000
3/3 [==============================] - 1s 361ms/step - loss: 1.2910 - categorical_ac
curacy: 0.7059
Epoch 347/2000
3/3 [==============================] - 1s 273ms/step - loss: 1.4599 - categorical_ac
curacy: 0.6353
Epoch 348/2000
3/3 [==============================] - 1s 301ms/step - loss: 1.7851 - categorical_ac
curacy: 0.5529
Epoch 349/2000
3/3 [==============================] - 1s 273ms/step - loss: 0.7091 - categorical_ac
curacy: 0.6706
Epoch 350/2000
3/3 [==============================] - 1s 276ms/step - loss: 0.6038 - categorical_ac
curacy: 0.7294
Epoch 351/2000
3/3 [==============================] - 1s 289ms/step - loss: 0.6587 - categorical_ac
curacy: 0.7412
Epoch 352/2000
3/3 [==============================] - 1s 335ms/step - loss: 0.5063 - categorical_ac
curacy: 0.8000
Epoch 353/2000
3/3 [==============================] - 1s 305ms/step - loss: 0.4860 - categorical_ac
curacy: 0.8118
Epoch 354/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.5285 - categorical_ac
curacy: 0.8118
Epoch 355/2000
3/3 [==============================] - 1s 308ms/step - loss: 0.4654 - categorical_ac
```

```
curacy: 0.8353
Epoch 356/2000
3/3 [==============================] - 1s 326ms/step - loss: 0.4886 - categorical_ac
curacy: 0.8000
Epoch 357/2000
3/3 [==============================] - 1s 270ms/step - loss: 0.5114 - categorical_ac
curacy: 0.7647
Epoch 358/2000
3/3 [==============================] - 1s 276ms/step - loss: 0.4529 - categorical_ac
curacy: 0.7765
Epoch 359/2000
3/3 [==============================] - 1s 303ms/step - loss: 0.5376 - categorical_ac
curacy: 0.7176
Epoch 360/2000
3/3 [==============================] - 1s 344ms/step - loss: 1.0074 - categorical_ac
curacy: 0.6706
Epoch 361/2000
3/3 [==============================] - 1s 250ms/step - loss: 0.5673 - categorical_ac
curacy: 0.7765
Epoch 362/2000
3/3 [==============================] - 1s 255ms/step - loss: 0.9367 - categorical_ac
curacy: 0.6235
Epoch 363/2000
3/3 [==============================] - 1s 260ms/step - loss: 0.9066 - categorical_ac
curacy: 0.6706
Epoch 364/2000
3/3 [==============================] - 1s 285ms/step - loss: 0.6478 - categorical_ac
curacy: 0.7412
Epoch 365/2000
3/3 [==============================] - 1s 263ms/step - loss: 0.8121 - categorical_ac
curacy: 0.7412
Epoch 366/2000
3/3 [==============================] - 1s 258ms/step - loss: 0.8696 - categorical_ac
curacy: 0.6706
Epoch 367/2000
3/3 [==============================] - 1s 291ms/step - loss: 0.7924 - categorical_ac
curacy: 0.7412
Epoch 368/2000
3/3 [==============================] - 1s 280ms/step - loss: 0.8385 - categorical_ac
curacy: 0.7412
Epoch 369/2000
3/3 [==============================] - 1s 278ms/step - loss: 0.5794 - categorical_ac
curacy: 0.7647
Epoch 370/2000
3/3 [==============================] - 1s 315ms/step - loss: 0.7631 - categorical_ac
curacy: 0.6235
Epoch 371/2000
3/3 [==============================] - 1s 262ms/step - loss: 0.8529 - categorical_ac
curacy: 0.6941
Epoch 372/2000
3/3 [==============================] - 1s 257ms/step - loss: 0.4603 - categorical_ac
curacy: 0.8118
Epoch 373/2000
3/3 [==============================] - 1s 353ms/step - loss: 0.8888 - categorical_ac
curacy: 0.6588
Epoch 374/2000
```

```
3/3 [==============================] - 1s 248ms/step - loss: 0.6470 - categorical_ac
curacy: 0.7529
Epoch 375/2000
3/3 [==============================] - 1s 331ms/step - loss: 0.8679 - categorical_ac
curacy: 0.7529
Epoch 376/2000
3/3 [==============================] - 1s 266ms/step - loss: 0.7238 - categorical_ac
curacy: 0.7765
Epoch 377/2000
3/3 [==============================] - 1s 257ms/step - loss: 0.5291 - categorical_ac
curacy: 0.7294
Epoch 378/2000
3/3 [==============================] - 1s 246ms/step - loss: 0.5834 - categorical_ac
curacy: 0.7882
Epoch 379/2000
3/3 [==============================] - 1s 290ms/step - loss: 0.7072 - categorical_ac
curacy: 0.7059
Epoch 380/2000
3/3 [==============================] - 1s 284ms/step - loss: 0.6241 - categorical_ac
curacy: 0.7294
Epoch 381/2000
3/3 [==============================] - 1s 241ms/step - loss: 0.3930 - categorical_ac
curacy: 0.8353
Epoch 382/2000
3/3 [==============================] - 1s 291ms/step - loss: 1.3545 - categorical_ac
curacy: 0.5882
Epoch 383/2000
3/3 [==============================] - 1s 281ms/step - loss: 1.5150 - categorical_ac
curacy: 0.4824
Epoch 384/2000
3/3 [==============================] - 1s 292ms/step - loss: 1.1039 - categorical_ac
curacy: 0.7176
Epoch 385/2000
3/3 [==============================] - 1s 340ms/step - loss: 0.9733 - categorical_ac
curacy: 0.7059
Epoch 386/2000
3/3 [==============================] - 1s 323ms/step - loss: 1.7427 - categorical_ac
curacy: 0.4941
Epoch 387/2000
3/3 [==============================] - 1s 325ms/step - loss: 1.4611 - categorical_ac
curacy: 0.5647
Epoch 388/2000
3/3 [==============================] - 1s 289ms/step - loss: 1.2645 - categorical_ac
curacy: 0.6000
Epoch 389/2000
3/3 [==============================] - 1s 261ms/step - loss: 0.8853 - categorical_ac
curacy: 0.6706
Epoch 390/2000
3/3 [==============================] - 1s 281ms/step - loss: 0.9785 - categorical_ac
curacy: 0.6471
Epoch 391/2000
3/3 [==============================] - 1s 277ms/step - loss: 0.8477 - categorical_ac
curacy: 0.6471
Epoch 392/2000
3/3 [==============================] - 1s 334ms/step - loss: 0.8472 - categorical_ac
curacy: 0.6588
```

```
Epoch 393/2000
3/3 [==============================] - 1s 337ms/step - loss: 0.8938 - categorical_ac
curacy: 0.6588
Epoch 394/2000
3/3 [==============================] - 1s 272ms/step - loss: 1.0787 - categorical_ac
curacy: 0.6941
Epoch 395/2000
3/3 [==============================] - 1s 266ms/step - loss: 0.9071 - categorical_ac
curacy: 0.6706
Epoch 396/2000
3/3 [==============================] - 1s 335ms/step - loss: 0.8873 - categorical_ac
curacy: 0.6941
Epoch 397/2000
3/3 [==============================] - 1s 298ms/step - loss: 0.9078 - categorical_ac
curacy: 0.7294
Epoch 398/2000
3/3 [==============================] - 1s 252ms/step - loss: 1.3685 - categorical_ac
curacy: 0.6588
Epoch 399/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.9628 - categorical_ac
curacy: 0.6706
Epoch 400/2000
3/3 [==============================] - 1s 274ms/step - loss: 0.7431 - categorical_ac
curacy: 0.7176
Epoch 401/2000
3/3 [==============================] - 1s 278ms/step - loss: 0.7268 - categorical_ac
curacy: 0.7176
Epoch 402/2000
3/3 [==============================] - 1s 297ms/step - loss: 0.6368 - categorical_ac
curacy: 0.7176
Epoch 403/2000
3/3 [==============================] - 1s 288ms/step - loss: 0.9850 - categorical_ac
curacy: 0.6824
Epoch 404/2000
3/3 [==============================] - 1s 252ms/step - loss: 1.0727 - categorical_ac
curacy: 0.6588
Epoch 405/2000
3/3 [==============================] - 1s 272ms/step - loss: 0.7926 - categorical_ac
curacy: 0.7412
Epoch 406/2000
3/3 [==============================] - 1s 370ms/step - loss: 0.4665 - categorical_ac
curacy: 0.7882
Epoch 407/2000
3/3 [==============================] - 1s 312ms/step - loss: 0.7020 - categorical_ac
curacy: 0.6824
Epoch 408/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.3908 - categorical_ac
curacy: 0.8118
Epoch 409/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.3586 - categorical_ac
curacy: 0.8235
Epoch 410/2000
3/3 [==============================] - 1s 265ms/step - loss: 0.6060 - categorical_ac
curacy: 0.7294
Epoch 411/2000
3/3 [==============================] - 1s 251ms/step - loss: 0.6034 - categorical_ac
```

```
curacy: 0.7294
Epoch 412/2000
3/3 [==============================] - 1s 273ms/step - loss: 0.6008 - categorical_ac
curacy: 0.7412
Epoch 413/2000
3/3 [==============================] - 1s 284ms/step - loss: 0.6008 - categorical_ac
curacy: 0.7059
Epoch 414/2000
3/3 [==============================] - 1s 258ms/step - loss: 0.5246 - categorical_ac
curacy: 0.7882
Epoch 415/2000
3/3 [==============================] - 1s 335ms/step - loss: 0.6184 - categorical_ac
curacy: 0.7412
Epoch 416/2000
3/3 [==============================] - 1s 277ms/step - loss: 0.6074 - categorical_ac
curacy: 0.7412
Epoch 417/2000
3/3 [==============================] - 1s 249ms/step - loss: 0.7238 - categorical_ac
curacy: 0.7412
Epoch 418/2000
3/3 [==============================] - 1s 269ms/step - loss: 1.0582 - categorical_ac
curacy: 0.6353
Epoch 419/2000
3/3 [==============================] - 1s 312ms/step - loss: 0.6336 - categorical_ac
curacy: 0.7294
Epoch 420/2000
3/3 [==============================] - 1s 280ms/step - loss: 1.0588 - categorical_ac
curacy: 0.6235
Epoch 421/2000
3/3 [==============================] - 1s 263ms/step - loss: 0.3797 - categorical_ac
curacy: 0.8353
Epoch 422/2000
3/3 [==============================] - 1s 315ms/step - loss: 0.4121 - categorical_ac
curacy: 0.7882
Epoch 423/2000
3/3 [==============================] - 1s 308ms/step - loss: 0.3191 - categorical_ac
curacy: 0.8353
Epoch 424/2000
3/3 [==============================] - 1s 346ms/step - loss: 0.3373 - categorical_ac
curacy: 0.8118
Epoch 425/2000
3/3 [==============================] - 1s 342ms/step - loss: 0.3195 - categorical_ac
curacy: 0.8235
Epoch 426/2000
3/3 [==============================] - 1s 299ms/step - loss: 0.3885 - categorical_ac
curacy: 0.8471
Epoch 427/2000
3/3 [==============================] - 1s 257ms/step - loss: 0.2763 - categorical_ac
curacy: 0.8941
Epoch 428/2000
3/3 [==============================] - 1s 260ms/step - loss: 0.3317 - categorical_ac
curacy: 0.8000
Epoch 429/2000
3/3 [==============================] - 1s 307ms/step - loss: 0.2865 - categorical_ac
curacy: 0.8706
Epoch 430/2000
```

```
3/3 [==============================] - 1s 258ms/step - loss: 0.3908 - categorical_ac
curacy: 0.7765
Epoch 431/2000
3/3 [==============================] - 1s 328ms/step - loss: 0.2527 - categorical_ac
curacy: 0.9176
Epoch 432/2000
3/3 [==============================] - 1s 332ms/step - loss: 0.2963 - categorical_ac
curacy: 0.8588
Epoch 433/2000
3/3 [==============================] - 1s 309ms/step - loss: 0.3560 - categorical_ac
curacy: 0.7882
Epoch 434/2000
3/3 [==============================] - 1s 272ms/step - loss: 0.4697 - categorical_ac
curacy: 0.7412
Epoch 435/2000
3/3 [==============================] - 1s 258ms/step - loss: 0.4920 - categorical_ac
curacy: 0.7412
Epoch 436/2000
3/3 [==============================] - 1s 333ms/step - loss: 0.4737 - categorical_ac
curacy: 0.7765
Epoch 437/2000
3/3 [==============================] - 1s 327ms/step - loss: 0.4862 - categorical_ac
curacy: 0.7647
Epoch 438/2000
3/3 [==============================] - 1s 254ms/step - loss: 0.5691 - categorical_ac
curacy: 0.7412
Epoch 439/2000
3/3 [==============================] - 1s 289ms/step - loss: 0.4481 - categorical_ac
curacy: 0.7765
Epoch 440/2000
3/3 [==============================] - 1s 324ms/step - loss: 0.2618 - categorical_ac
curacy: 0.8824
Epoch 441/2000
3/3 [==============================] - 1s 255ms/step - loss: 0.2682 - categorical_ac
curacy: 0.8824
Epoch 442/2000
3/3 [==============================] - 1s 256ms/step - loss: 0.3168 - categorical_ac
curacy: 0.8706
Epoch 443/2000
3/3 [==============================] - 1s 373ms/step - loss: 0.2596 - categorical_ac
curacy: 0.9294
Epoch 444/2000
3/3 [==============================] - 1s 312ms/step - loss: 0.2722 - categorical_ac
curacy: 0.8824
Epoch 445/2000
3/3 [==============================] - 1s 267ms/step - loss: 0.3002 - categorical_ac
curacy: 0.8353
Epoch 446/2000
2/3 [==================>.....] - ETA: 0s - loss: 0.3097 - categorical_accurac
y: 0.8750
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[43], line 1
----> 1 model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback])

File ~\.conda\envs\py310\lib\site-packages\keras\utils\traceback_utils.py:65, in fil
ter_traceback.<locals>.error_handler(*args, **kwargs)
     63 filtered_tb = None
     64 try:
---> 65     return fn(*args, **kwargs)
     66 except Exception as e:
     67     filtered_tb = _process_traceback_frames(e.__traceback__)

File ~\.conda\envs\py310\lib\site-packages\keras\engine\training.py:1564, in Model.f
it(self, x, y, batch_size, epochs, verbose, callbacks, validation_split, validation_
data, shuffle, class_weight, sample_weight, initial_epoch, steps_per_epoch, validati
on_steps, validation_batch_size, validation_freq, max_queue_size, workers, use_multi
processing)
   1556 with tf.profiler.experimental.Trace(
   1557     "train",
   1558     epoch_num=epoch,
   (...)
   1561     _r=1,
   1562 ):
   1563     callbacks.on_train_batch_begin(step)
-> 1564     tmp_logs = self.train_function(iterator)
   1565     if data_handler.should_sync:
   1566         context.async_wait()

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\util\traceback_utils.p
y:150, in filter_traceback.<locals>.error_handler(*args, **kwargs)
    148 filtered_tb = None
    149 try:
--> 150   return fn(*args, **kwargs)
    151 except Exception as e:
    152   filtered_tb = _process_traceback_frames(e.__traceback__)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\def_function.py:9
15, in Function.__call__(self, *args, **kwds)
    912 compiler = "xla" if self._jit_compile else "nonXla"
    914 with OptionalXlaContext(self._jit_compile):
--> 915   result = self._call(*args, **kwds)
    917 new_tracing_count = self.experimental_get_tracing_count()
    918 without_tracing = (tracing_count == new_tracing_count)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\def_function.py:9
47, in Function._call(self, *args, **kwds)
    944   self._lock.release()
    945   # In this case we have created variables on the first call, so we run the
    946   # defunned version which is guaranteed to never create variables.
--> 947   return self._stateless_fn(*args, **kwds)  # pylint: disable=not-callable
    948 elif self._stateful_fn is not None:
    949   # Release the lock early so that multiple threads can perform the call
    950   # in parallel.
    951   self._lock.release()
```

```
File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\function.py:2496,
in Function.__call__(self, *args, **kwargs)
   2493 with self._lock:
   2494   (graph_function,
   2495    filtered_flat_args) = self._maybe_define_function(args, kwargs)
-> 2496 return graph_function._call_flat(
   2497     filtered_flat_args, captured_inputs=graph_function.captured_inputs)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\function.py:1862,
in ConcreteFunction._call_flat(self, args, captured_inputs, cancellation_manager)
   1858 possible_gradient_type = gradients_util.PossibleTapeGradientTypes(args)
   1859 if (possible_gradient_type == gradients_util.POSSIBLE_GRADIENT_TYPES_NONE
   1860     and executing_eagerly):
   1861   # No tape is watching; skip to running the function.
-> 1862   return self._build_call_outputs(self._inference_function.call(
   1863       ctx, args, cancellation_manager=cancellation_manager))
   1864 forward_backward = self._select_forward_and_backward_functions(
   1865     args,
   1866     possible_gradient_type,
   1867     executing_eagerly)
   1868 forward_function, args_with_tangents = forward_backward.forward()

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\function.py:499,
in _EagerDefinedFunction.call(self, ctx, args, cancellation_manager)
    497 with _InterpolateFunctionError(self):
    498   if cancellation_manager is None:
--> 499     outputs = execute.execute(
    500         str(self.signature.name),
    501         num_outputs=self._num_outputs,
    502         inputs=args,
    503         attrs=attrs,
    504         ctx=ctx)
    505   else:
    506     outputs = execute.execute_with_cancellation(
    507         str(self.signature.name),
    508         num_outputs=self._num_outputs,
   (...)
    511         ctx=ctx,
    512         cancellation_manager=cancellation_manager)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\execute.py:54, in
quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     52 try:
     53   ctx.ensure_initialized()
---> 54   tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
     55                                       inputs, attrs, num_outputs)
     56 except core._NotOkStatusException as e:
     57   if name is not None:

KeyboardInterrupt:
```

In [27]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 30, 64)            442112
_________________________________________________________________
lstm_1 (LSTM)                (None, 30, 128)           98816
_________________________________________________________________
lstm_2 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense (Dense)                (None, 64)                4160
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 99
=================================================================
Total params: 596,675
Trainable params: 596,675
Non-trainable params: 0
_________________________________________________________________
```

Make Predictions

In [44]:
```
res = model.predict(X_test)
```

```
1/1 [==============================] - 0s 399ms/step
```

In [45]:
```
actions[np.argmax(res[4])]
```

Out[45]: 'thanks'

In [46]:
```
actions[np.argmax(y_test[4])]
```

Out[46]: 'thanks'

Save Weights

In [48]:
```
model.save(r'D:\data science\Practice\asl\signlangmodel.h5')
```

In [10]:
```
from tensorflow.keras.models import load_model
model = load_model(r'D:\data science\Practice\asl\signlangmodel.h5')
```

```
WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the c
riteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_1 will not use cuDNN kernels since it doesn't meet the
criteria. It will use a generic GPU kernel as fallback when running on GPU.
WARNING:tensorflow:Layer lstm_2 will not use cuDNN kernels since it doesn't meet the
criteria. It will use a generic GPU kernel as fallback when running on GPU.
```

Evaluation using Confusion Matrix and Accuracy

In [50]:
```
from sklearn.metrics import multilabel_confusion_matrix, accuracy_score
```

In [51]:
```
yhat = model.predict(X_test)
```

```
1/1 [==============================] - 0s 66ms/step
```

In [52]:
```
ytrue = np.argmax(y_test, axis=1).tolist()
yhat = np.argmax(yhat, axis=1).tolist()
```

In [53]:
```
multilabel_confusion_matrix(ytrue, yhat)
```

Out[53]:
```
array([[[3, 0],
        [0, 2]],

       [[4, 0],
        [0, 1]],

       [[3, 0],
        [0, 2]]], dtype=int64)
```

In [54]:
```
accuracy_score(ytrue, yhat)
```

Out[54]:
```
1.0
```

Class text to Audio

In [23]:
```
import cv2
import numpy as np
import mediapipe as mp
import pyttsx3
import time  # Import time module for delay

# Initialize text-to-speech engine
engine = pyttsx3.init()

# New detection variables
sequence = []
sentence = []
predictions = []
threshold = 0.5
prev_prediction = None  # Variable to store the previous prediction

# Actions that we try to detect
actions = ['hello', 'thanks', 'iloveyou']

# Set up VideoCapture
cap = cv2.VideoCapture(0)

# Set mediapipe model
with mp.solutions.holistic.Holistic(min_detection_confidence=0.5, min_tracking_conf
    while cap.isOpened():
        # Read feed
        ret, frame = cap.read()
```

```
        # Make detections
        image, results = mediapipe_detection(frame, holistic)

        # Prediction logic
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]

        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            predicted_class = np.argmax(res)
            print(actions[predicted_class])
            predictions.append(predicted_class)

            # TTS logic for predicted classes
            if res[predicted_class] > threshold and predicted_class != prev_predict
                predicted_action = actions[predicted_class]
                engine.say(predicted_action)
                engine.runAndWait()
                prev_prediction = predicted_class  # Update previous prediction
                time.sleep(1)  # Add a 1-second delay between predictions

        # Display the frame
        cv2.imshow('OpenCV Feed', image)

        # Break gracefully if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

# Release the capture and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 47ms/step
thanks
1/1 [==============================] - 0s 81ms/step
thanks
1/1 [==============================] - 0s 61ms/step
thanks
1/1 [==============================] - 0s 78ms/step
thanks
1/1 [==============================] - 0s 62ms/step
thanks
1/1 [==============================] - 0s 67ms/step
hello
1/1 [==============================] - 0s 64ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 64ms/step
hello
1/1 [==============================] - 0s 64ms/step
hello
1/1 [==============================] - 0s 64ms/step
hello
1/1 [==============================] - 0s 56ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 63ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 55ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 58ms/step
thanks
1/1 [==============================] - 0s 58ms/step
hello
1/1 [==============================] - 0s 58ms/step
thanks
1/1 [==============================] - 0s 56ms/step
thanks
1/1 [==============================] - 0s 57ms/step
hello
```

In [55]:
```
from scipy import stats
```

In [56]:
```
colors = [(245,117,16), (117,245,16), (16,117,245)]
def prob_viz(res, actions, input_frame, colors):
    output_frame = input_frame.copy()
    for num, prob in enumerate(res):
        cv2.rectangle(output_frame, (0,60+num*40), (int(prob*100), 90+num*40), colo
        cv2.putText(output_frame, actions[num], (0, 85+num*40), cv2.FONT_HERSHEY_SI

    return output_frame
```

In [58]:

```
# 1. New detection variables
sequence = []
sentence = []
predictions = []
threshold = 0.5

cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5
    while cap.isOpened():

        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)
        print(results)

        # Draw landmarks
        draw_styled_landmarks(image, results)

        # 2. Prediction logic
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]

        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            print(actions[np.argmax(res)])
            predictions.append(np.argmax(res))


        #3. Viz logic
            if np.unique(predictions[-10:])[0]==np.argmax(res):
                if res[np.argmax(res)] > threshold:

                    if len(sentence) > 0:
                        if actions[np.argmax(res)] != sentence[-1]:
                            sentence.append(actions[np.argmax(res)])
                    else:
                        sentence.append(actions[np.argmax(res)])

            if len(sentence) > 5:
                sentence = sentence[-5:]

            # Viz probabilities
            image = prob_viz(res, actions, image, colors)

        cv2.rectangle(image, (0,0), (640, 40), (245, 117, 16), -1)
        cv2.putText(image, ' '.join(sentence), (3,30),
                       cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA

        # Show to screen
        cv2.imshow('OpenCV Feed', image)
```

```
        # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

```
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 58ms/step
iloveyou
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 50ms/step
iloveyou
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 71ms/step
iloveyou
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 58ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 62ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
thanks
```

```
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 63ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 68ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 63ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 54ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 52ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 65ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 82ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 65ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 63ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 55ms/step
```

```
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 64ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 72ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 65ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 55ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 62ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 58ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 67ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 65ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 68ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
```

```
1/1 [==============================] - 0s 65ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 73ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 58ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 52ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 69ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 65ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 53ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 74ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 78ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 54ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 54ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 54ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 76ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 63ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 63ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
hello
```

```
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 107ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 64ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
thanks
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 58ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 67ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 55ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 66ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 57ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 64ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 59ms/step
iloveyou
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 63ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 54ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 64ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 53ms/step
```

```
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 64ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 62ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 62ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 70ms/step
iloveyou
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 74ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 61ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 70ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 64ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 62ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 58ms/step
thanks
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 60ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 65ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 66ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
```

```
1/1 [==============================] - 0s 53ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 66ms/step
hello
<class 'mediapipe.python.solution_base.SolutionOutputs'>
1/1 [==============================] - 0s 56ms/step
hello
```

In [ ]:

```python
import cv2
import numpy as np
import mediapipe as mp
import pyttsx3
import time  # Import time module for delay

# Initialize text-to-speech engine
engine = pyttsx3.init()

# New detection variables
sequence = []
sentence = []
predictions = []
threshold = 0.5
prev_prediction = None  # Variable to store the previous prediction

# Actions that we try to detect
actions = ['hello', 'thanks', 'iloveyou']

# Set up VideoCapture
cap = cv2.VideoCapture(0)

# Set mediapipe model
with mp.solutions.holistic.Holistic(min_detection_confidence=0.5, min_tracking_conf
    while cap.isOpened():
        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)

        # Prediction logic
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]

        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            predicted_class = np.argmax(res)
            print(actions[predicted_class])
            predictions.append(predicted_class)

            # TTS logic for predicted classes
            if res[predicted_class] > threshold and predicted_class != prev_predict
                predicted_action = actions[predicted_class]
                engine.say(predicted_action)
                engine.runAndWait()
```

```
                prev_prediction = predicted_class  # Update previous prediction
                time.sleep(3)  # Add a 1-second delay between predictions

        # Display the frame
        cv2.imshow('OpenCV Feed', image)

        # Break gracefully if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

# Release the capture and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

In [21]:
```
cap.release()
cv2.destroyAllWindows()
```

In [14]:
```
import cv2
import numpy as np
import mediapipe as mp
import pyttsx3

# Initialize text-to-speech engine
engine = pyttsx3.init()

# New detection variables
sequence = []
sentence = []
predictions = []
threshold = 0.5
prev_prediction = None  # Variable to store the previous prediction

# Actions that we try to detect
actions = ['hello', 'thanks', 'iloveyou']

# Set up VideoCapture
cap = cv2.VideoCapture(0)

# Set mediapipe model
with mp.solutions.holistic.Holistic(min_detection_confidence=0.5, min_tracking_conf
    while cap.isOpened():
        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)

        # Prediction logic
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]

        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            predicted_class = np.argmax(res)
```

```
            predicted_action = actions[predicted_class]
            print(predicted_action)  # Print the class name to the console
            predictions.append(predicted_class)

            # TTS logic for predicted classes
            if res[predicted_class] > threshold and predicted_class != prev_predict
                engine.say(predicted_action)
                engine.runAndWait()
                prev_prediction = predicted_class  # Update previous prediction

        # Break gracefully if 'q' is pressed
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

# Release the capture and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

```
1/1 [==============================] - 0s 47ms/step
thanks
1/1 [==============================] - 0s 62ms/step
thanks
1/1 [==============================] - 0s 58ms/step
thanks
1/1 [==============================] - 0s 58ms/step
thanks
1/1 [==============================] - 0s 58ms/step
thanks
1/1 [==============================] - 0s 58ms/step
thanks
1/1 [==============================] - 0s 61ms/step
thanks
1/1 [==============================] - 0s 69ms/step
thanks
1/1 [==============================] - 0s 59ms/step
thanks
1/1 [==============================] - 0s 56ms/step
thanks
1/1 [==============================] - 0s 57ms/step
thanks
1/1 [==============================] - 0s 57ms/step
thanks
1/1 [==============================] - 0s 56ms/step
thanks
1/1 [==============================] - 0s 57ms/step
thanks
1/1 [==============================] - 0s 58ms/step
hello
Exception ignored in: <function BSTR.__del__ at 0x000001BD6B931240>
Traceback (most recent call last):
  File "C:\Users\deepchanddc2\.conda\envs\py310\lib\site-packages\comtypes\__init__.
py", line 683, in __del__
    def __del__(self, _free=windll.oleaut32.SysFreeString):
KeyboardInterrupt:
```

```
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 56ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 62ms/step
hello
1/1 [==============================] - 0s 63ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 61ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 54ms/step
hello
1/1 [==============================] - 0s 65ms/step
thanks
1/1 [==============================] - 0s 58ms/step
hello
1/1 [==============================] - 0s 55ms/step
hello
1/1 [==============================] - 0s 60ms/step
hello
1/1 [==============================] - 0s 55ms/step
hello
1/1 [==============================] - 0s 61ms/step
hello
1/1 [==============================] - 0s 72ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 63ms/step
hello
1/1 [==============================] - 0s 61ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 55ms/step
hello
1/1 [==============================] - 0s 58ms/step
hello
1/1 [==============================] - 0s 64ms/step
hello
1/1 [==============================] - 0s 54ms/step
hello
1/1 [==============================] - 0s 56ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
```

```
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 58ms/step
hello
1/1 [==============================] - 0s 60ms/step
hello
1/1 [==============================] - 0s 61ms/step
hello
1/1 [==============================] - 0s 54ms/step
hello
1/1 [==============================] - 0s 58ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 60ms/step
hello
1/1 [==============================] - 0s 56ms/step
hello
1/1 [==============================] - 0s 59ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 57ms/step
hello
1/1 [==============================] - 0s 58ms/step
hello
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[14], line 37
     34 sequence = sequence[-30:]
     36 if len(sequence) == 30:
---> 37     res = model.predict(np.expand_dims(sequence, axis=0))[0]
     38     predicted_class = np.argmax(res)
     39     predicted_action = actions[predicted_class]

File ~\.conda\envs\py310\lib\site-packages\keras\utils\traceback_utils.py:65, in fil
ter_traceback.<locals>.error_handler(*args, **kwargs)
     63 filtered_tb = None
     64 try:
---> 65     return fn(*args, **kwargs)
     66 except Exception as e:
     67     filtered_tb = _process_traceback_frames(e.__traceback__)

File ~\.conda\envs\py310\lib\site-packages\keras\engine\training.py:2253, in Model.p
redict(self, x, batch_size, verbose, steps, callbacks, max_queue_size, workers, use_
multiprocessing)
   2251 for step in data_handler.steps():
   2252     callbacks.on_predict_batch_begin(step)
-> 2253     tmp_batch_outputs = self.predict_function(iterator)
   2254     if data_handler.should_sync:
   2255         context.async_wait()

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\util\traceback_utils.p
y:150, in filter_traceback.<locals>.error_handler(*args, **kwargs)
    148 filtered_tb = None
    149 try:
--> 150   return fn(*args, **kwargs)
    151 except Exception as e:
    152   filtered_tb = _process_traceback_frames(e.__traceback__)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\def_function.py:9
15, in Function.__call__(self, *args, **kwds)
    912 compiler = "xla" if self._jit_compile else "nonXla"
    914 with OptionalXlaContext(self._jit_compile):
--> 915   result = self._call(*args, **kwds)
    917 new_tracing_count = self.experimental_get_tracing_count()
    918 without_tracing = (tracing_count == new_tracing_count)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\def_function.py:9
54, in Function._call(self, *args, **kwds)
    951 self._lock.release()
    952 # In this case we have not created variables on the first call. So we can
    953 # run the first trace but we should fail if variables are created.
--> 954 results = self._stateful_fn(*args, **kwds)
    955 if self._created_variables and not ALLOW_DYNAMIC_VARIABLE_CREATION:
    956   raise ValueError("Creating variables on a non-first call to a function"
    957                    " decorated with tf.function.")

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\function.py:2496,
in Function.__call__(self, *args, **kwargs)
   2493 with self._lock:
   2494   (graph_function,
```

```
   2495     filtered_flat_args) = self._maybe_define_function(args, kwargs)
-> 2496 return graph_function._call_flat(
   2497     filtered_flat_args, captured_inputs=graph_function.captured_inputs)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\function.py:1862,
in ConcreteFunction._call_flat(self, args, captured_inputs, cancellation_manager)
   1858 possible_gradient_type = gradients_util.PossibleTapeGradientTypes(args)
   1859 if (possible_gradient_type == gradients_util.POSSIBLE_GRADIENT_TYPES_NONE
   1860     and executing_eagerly):
   1861   # No tape is watching; skip to running the function.
-> 1862   return self._build_call_outputs(self._inference_function.call(
   1863       ctx, args, cancellation_manager=cancellation_manager))
   1864 forward_backward = self._select_forward_and_backward_functions(
   1865     args,
   1866     possible_gradient_type,
   1867     executing_eagerly)
   1868 forward_function, args_with_tangents = forward_backward.forward()

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\function.py:499,
in _EagerDefinedFunction.call(self, ctx, args, cancellation_manager)
    497 with _InterpolateFunctionError(self):
    498   if cancellation_manager is None:
--> 499     outputs = execute.execute(
    500         str(self.signature.name),
    501         num_outputs=self._num_outputs,
    502         inputs=args,
    503         attrs=attrs,
    504         ctx=ctx)
    505   else:
    506     outputs = execute.execute_with_cancellation(
    507         str(self.signature.name),
    508         num_outputs=self._num_outputs,
   (...)
    511         ctx=ctx,
    512         cancellation_manager=cancellation_manager)

File ~\.conda\envs\py310\lib\site-packages\tensorflow\python\eager\execute.py:54, in
quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     52 try:
     53   ctx.ensure_initialized()
---> 54   tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
     55                                       inputs, attrs, num_outputs)
     56 except core._NotOkStatusException as e:
     57   if name is not None:

KeyboardInterrupt:
```

In [ ]:

In [ ]: